A Survey of Unsupervised Deep Domain Adaptation

GARRETT WILSON and DIANE J. COOK, Washington State University, USA

Deep learning has produced state-of-the-art results for a variety of tasks. While such approaches for supervised learning have performed well, they assume that training and testing data are drawn from the same distribution, which may not always be the case. As a complement to this challenge, unsupervised domain adaptation can handle situations where a network is trained on labeled data from a source domain and unlabeled data from a related but different target domain with the goal of performing well at test-time on the target domain. Many unsupervised deep domain adaptation approaches have thus been developed. This survey will compare these approaches by examining alternative methods, the unique and common elements, results, and theoretical insights. We follow this with a look at application areas and open research directions.

1 INTRODUCTION

Supervised learning is arguably the most prevalent type of machine learning and has enjoyed much success across diverse application areas. However, many supervised learning methods make a common assumption: the training and testing data are drawn from the same distribution. When this constraint is violated, a classifier trained on the *source domain* will likely experience a drop in performance when tested on the *target domain* due to the differences between domains [158]. *Domain adaptation* refers to the goal of learning a concept from labeled data in a source domain that performs well on a different but related target domain [61, 68, 156]. *Unsupervised* domain adaptation specifically addresses the situation where there is labeled source data and only unlabeled target data available for use during training [61, 126].

Because of its ability to adapt labeled data for use in a new application, domain adaptation can reduce the need for costly labeled data in the target domain. As an example, consider the problem of semantically segmenting images. Each real image in the Cityscapes dataset required approximately 1.5 hours to annotate for semantic segmentation [39]. In this case, human annotation time could be spared by training an image semantic segmentation model on synthetic street view images (the source domain) since these can be cheaply generated, then adapting and testing for real street view images (the target domain, here the Cityscapes dataset).

An undeniable trend in machine learning is the increased usage of deep neural networks. Deep networks have produced many state-of-the-art results for a variety of machine learning tasks [61, 68] such as image classification, speech recognition, machine translation, and image generation [67, 68]. When trained on large amounts of data, these many-layer neural networks can learn powerful, hierarchical representations [68, 126, 198] and can be highly scalable [64]. At the same time, these networks can also experience performance drops due to domain shifts [60, 198]. Thus, much research has gone into adapting such networks from large labeled datasets to domains where little (or possibly no) labeled training data is available (for a list, see [223]). These unsupervised deep domain adaptation approaches, which combine the benefit of deep learning with the very practical use of domain adaptation to remove the reliance on potentially costly target data labels, will be the focus of this survey.

A number of surveys have been created on the topic of domain adaptation [10, 21, 36, 41, 42, 106, 139, 158, 199, 213, 246] and more generally transfer learning [38, 111, 131, 156, 189, 203, 205, 219, 234], of which domain adaptation can be viewed as a special case [158]. Previous domain adaptation surveys lack depth of coverage and comparison of unsupervised deep domain adaptation approaches.

Authors' address: Garrett Wilson, garrett.wilson@wsu.edu; Diane J. Cook, djcook@wsu.edu, Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA, 99164, USA.

In some cases, prior surveys do not discuss domain mapping [41, 42, 106], normalization statistic-based [41, 42, 106, 246], or ensemble-based [41, 42, 106, 213, 246] methods. In other cases, they do not survey deep learning approaches [10, 139, 158]. Still others are application-centric, focusing on a single use case such as machine translation [21, 36]. One earlier survey focuses on the multi-source scenario [199], while we focus on the more prevalent single-source scenario. Transfer learning is a broader topic to cover, thus surveys provide minimal coverage and comparison of the deep learning methods that have been designed for unsupervised domain adaptation [131, 156, 189, 203, 219, 234], or they focus on tasks such as activity recognition [38] or reinforcement learning [111, 205]. The goal of this survey is to discuss, highlight unique components, and compare approaches to unsupervised deep domain adaptation.

We first provide background on where domain adaptation fits into the more general problem of transfer learning. We follow this with an overview of generative adversarial networks (GANs) to provide background for the increasingly widespread use of adversarial techniques in domain adaptation. Next, we investigate the various domain adaptation methods, the components of those methods, and the results. Then, we overview domain adaptation theory and discuss what we can learn from the theoretical results. Finally, we look at application areas and identify future research directions for domain adaptation.

2 BACKGROUND

2.1 Transfer Learning

The focus of this survey is domain adaptation. Because domain adaptation can be viewed as a special case of transfer learning [158], we first review transfer learning to highlight the role of domain adaptation within this topic. Transfer learning is defined as the learning scenario where a model is trained on a source domain or task and evaluated on a different but related target domain or task, where either the tasks or domains (or both) differ [50, 68, 156, 219]. For instance, we may wish to learn a model on a handwritten digit dataset (e.g., MNIST [113]) with the goal of using it to recognize house numbers (e.g., SVHN [151]). Or, we may wish to learn a model on a synthetic, cheap-to-generate traffic sign dataset [145] with the goal of using it to classify real traffic signs (e.g., GTSRB [196]). In these examples, the source dataset used to train the model is related but different from the target dataset used to test the model – both are digits and signs respectively, but each dataset looks significantly different. When the source and target differ but are related, then transfer learning can be applied to obtain higher accuracy on the target data.

2.1.1 Categorizing Methods. In a transfer learning survey paper, Pan et al. [156] defined two terms to help classify various transfer learning techniques: “domain” and “task.” A domain consists of a feature space and a marginal probability distribution (i.e., the features of the data and the distribution of those features in the dataset). A task consists of a label space and an objective predictive function (i.e., the set of labels and a predictive function that is learned from the training data). Thus, a transfer learning problem might be either transferring knowledge from a source domain to a different target domain or transferring knowledge from a source task to a different target task (or a combination of the two) [50, 156, 219].

By this definition, a change in domain may result from either a change in feature space or a change in the marginal probability distribution. When classifying documents using text mining, a change in the feature space may result from a change in language (e.g., English to Spanish), whereas a change in the marginal probability distribution may result from a change in document topics (e.g., computer science to English literature) [156]. Similarly, a change in task may result from either a change in the label space or a change in the objective predictive function. In the case of document classification, a change in the label space may result from a change in the number of classes (e.g., from a set of 10 topic labels to a set of 100 topic labels). Similarly, a change in the objective predictive function may result from a substantial change in the distribution of the labels (e.g., the source domain has 100 instances of class A and 10,000 of class B, whereas the target has 10,000 instances of A and 100 of B) [156].

To classify transfer learning algorithms based on whether the task or domain differs between source and target, Pan et al. [156] introduced three terms: "inductive", "transductive", and "unsupervised" transfer learning. In inductive transfer learning, the target and source tasks are different, the domains may or may not differ, and some labeled target data is required. In transductive transfer learning, the tasks remain the same while the domains are different, and both labeled source data and unlabeled target data are required. Finally, in unsupervised transfer learning, the tasks differ as in the inductive case, but there is no requirement of labeled data in either the source domain or the target domain.

2.1.2 Domain Adaptation. One popular type of transfer learning is *domain adaptation*, which will be the focus of our survey. Domain adaptation is a type of transductive transfer learning. Here, the target task remains the same as the source, as well as the domain feature space, but the domain marginal probability distributions differ [156, 165]. Only part of the domain changes since the feature space is required to remain fixed between source and target.

In addition to the previous terminology, machine learning techniques are often categorized based on whether or not labeled training data is available. Supervised learning assumes labeled data is available, semi-supervised learning uses both labeled data and unlabeled data, and unsupervised learning uses only unlabeled data. However, domain adaptation assumes data comes from both a source domain and a target domain. Thus, prepending one of these three terms to "domain adaptation" is ambiguous since it may refer to labeled data being available in the source or target domains.

Authors apply these terms in various ways to domain adaptation [44, 96, 156, 178, 219]. In this paper, we will refer to "unsupervised" domain adaptation as the case in which both labeled source data and unlabeled target data are available, "semi-supervised" domain adaptation as the case in which labeled source data in addition to some labeled target data are available, and "supervised" domain adaptation as the case in which both labeled source and target data are available [10]. The distinction between these categories describes the target domain, but only describe situations in which labeled data is available for the source domain. These definitions are commonly used in the methods surveyed in this paper as well as others [23, 61, 64, 126, 178, 198].

2.1.3 Related Problems. Multi-domain learning [50, 98] and multi-task learning [25] are related to transfer learning and domain adaptation. In contrast to transfer learning, the goal of these learning approaches is obtaining high performance on all specified domains (or tasks) rather than just on a single target domain (or task) [156, 225]. For example, often it is assumed that the training data are drawn in an independent and identically distributed (i.i.d.) fashion, which may not be the case [98]. One such example is the task of developing a spam filter for users who disagree on what is considered spam. If all the users' data are combined, the training data will be drawn from multiple domains. While each individual domain may be i.i.d., the aggregated dataset may not be. If the data is split by user, then there may be too little data to learn a model for each user. Multi-domain learning can take advantage of the entire dataset to learn individual user preferences [50, 98]. Some researchers have developed adversarial strategies to tackle this multi-domain learning challenge [76, 187].

When working with multiple tasks, instead of training models separately for different tasks (e.g., one model for detecting shapes in an image and one model for detecting text in an image), multi-task learning will learn these separate but related tasks simultaneously so that they can

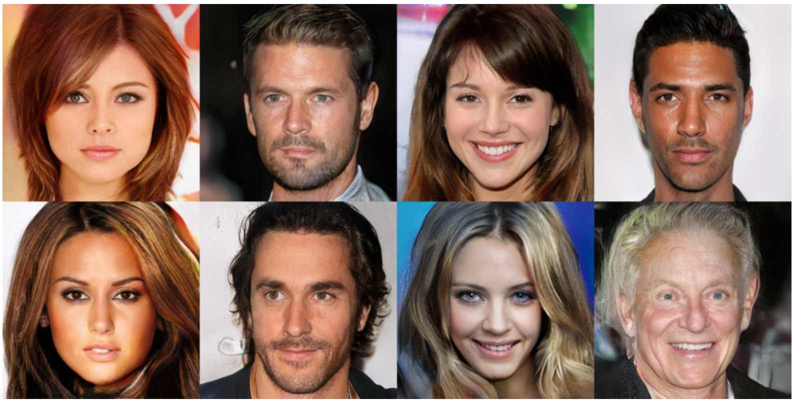


Fig. 1. Realistic but entirely synthetic images of human faces generated by a GAN trained on the CelebA-HQ dataset [101].

mutually benefit from the training data of other tasks through a (partially) shared representation [25]. If there are both multiple tasks and domains, then these approaches can be combined into multi-domain multi-task learning, as is described by Yang et al. [225].

Another related problem is domain generalization, in which a model is trained on multiple source domains with labeled data and then tested on a separate target domain that was not seen during training [150]. This contrasts with domain adaptation where target examples (possibly unlabeled) are available during training. Some approaches related to those surveyed in this paper have been designed to address this situation. Examples include an adversarial method introduced by Zhao et al. [245] and an autoencoder approach by Ghifary et al. [63] discussed in Section 7.4.

2.2 Generative Adversarial Networks

Many deep domain adaptation methods that we will discuss in the next section incorporate adversarial training. One popular use of adversarial training is generative adversarial networks (GANs). GANs have been directly incorporated into adversarial domain mapping methods (Section 3.2) and have inspired similar adversarial training setups in adversarial domain-invariant feature learning methods (Section 3.1.3). Thus, to provide background for these techniques, we will first discuss GANs.

In recent years there has been a large and growing interest in GANs. Pitting two well-matched neural networks against each other (hence "adversarial"), playing the roles of a data discriminator and a data generator, the pair is able to refine each player's abilities in order to perform functions such as synthetic data generation. Goodfellow et al. [69] proposed this technique in 2014. Since that time, hundreds of papers have been published on the topic [78, 233]. GANs have traditionally been applied to synthetic image generation, but recently researchers have been exploring other novel use cases such as domain adaptation.

GANs are a type of deep generative model [69]. For synthetic image generation, a training dataset of images must be available. Popular datasets include human faces (CelebA [125]), handwritten digits (MNIST [113]), bedrooms (LSUN [230]), and sets of other objects (CIFAR-10 [107] and ImageNet [45, 175]). After training, the generative model will be able to generate synthetic images that resemble those in the training data. For example, a generator trained with CelebA will generate images of human faces that look realistic but are not images of real people, as shown in Figure 1. To learn to do this, GANs utilize two neural networks competing against each other [69]. One network

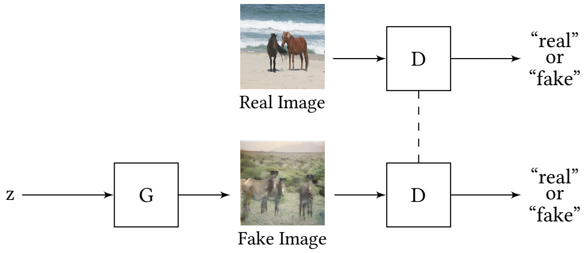


Fig. 2. Illustration of the GAN generator G and discriminator D networks. The dashed line between the D networks indicates that they share weights (or are the same network). In the top row, a real image from the training data (horses ↔ zebras dataset by Zhu et al. [248]) is fed to the discriminator, and the goal of D is to make $D(x) = 1$ (correctly classify as real). In the bottom row, a fake image from the generator is fed to the discriminator, and the goal of D is to make $D(G(z)) = 0$ (correctly classify as fake), which competes with the goal of G to make $D(G(z)) = 1$ (misclassify as real).

represents a generator. The generator accepts a noise vector as input, which contains random values drawn from some distribution such as normal or uniform. The goal of the generator network is to output a vector that is indistinguishable from the real training data. The other network represents a discriminator, which accepts as input either a real sample from the training data or a fake sample from the generator. The goal of the discriminator is to determine the probability that the input sample is real. During training, these two networks play a minimax game, where the generator tries to fool the discriminator and the discriminator attempts to not be fooled.

Using the notation from Goodfellow et al. [69], we define a value function $V(G, D)$ employed by the minimax game between the two networks:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \tag{1}$$

Here, $x \sim p_{data}(x)$ draws a sample from the real data distribution, $z \sim p_z(z)$ draws a sample from the input noise, $D(x; \theta_d)$ is the discriminator, and $G(z; \theta_g)$ is the generator. As shown in the equation, the goal is to find the parameters θ_d that maximize the log probability of correctly discriminating between real (x) and fake ($G(z)$) samples while at the same time finding the parameters θ_g that minimize the log probability of $1 - D(G(z))$. The term $D(G(z))$ represents the probability that generated data $G(z)$ is real. If the discriminator correctly classifies a fake input then $D(G(z)) = 0$. Equation 1 minimizes the quantity $1 - D(G(z))$. This occurs when $D(G(z)) = 1$, or when the discriminator misclassifies the generator's output as a real sample. Thus the discriminator's mission is to learn to correctly classify the input as real or fake while the generator tries to fool the discriminator into thinking that its generated output is real. This process is illustrated in Figure 2.

2.2.1 Training. In recent years there have been impressive results from GANs. At the same time, this research faces numerous challenges. Training a GAN can encounter problems such as difficulty converging, mode collapse, and vanishing gradients.

GAN training may fail to converge. Because there are two players in the GAN game, each player's move (i.e., update its neural network via gradient descent) toward a lower loss may undo the other player's progress toward reaching its lower loss [67]. For example, GANs have been observed to oscillate without making progress toward an equilibrium [67]. In general, an equilibrium to a game may not even exist (e.g., rock-paper-scissors [5]), but Arora et al. [5] show that an approximate pure equilibrium does exist for a Wasserstein training objective if the generator wins the game.

However, while an approximate equilibrium does exist, that does not mean that backpropagation will find it when training the GAN [5].

A common type of non-convergence that GANs may suffer is *mode collapse*, where the generator only learns to generate realistic samples for a few specialized modes of the data distribution [67]. For example, a generator may learn to only generate images of a particular type of dog when the dataset contains images of many different types of animals [67].

Another problem is *vanishing gradients*. A solution to the minimax game is found through iterative optimization: alternating between optimizing the discriminator objective and the generator objective. When the generated samples are initially very poor, however, the discriminator will be confident in whether the generated image is real or fake. Thus, $D(G(z))$, which is the probability of the generated sample being real, will be close to zero, causing the gradient of $\log(1 - D(G(z))$ to be small [69].

Many methods have been proposed to resolve these training challenges. Even in the original GAN paper, Goodfellow et al. [69] proposed a variation of the objective in Equation 1, replacing minimizing $\log(1 - D(\tilde{\mathbf{x}}))$ with maximizing $\log(D(\tilde{\mathbf{x}}))$ to reduce problems from vanishing gradients (referred to as a non-saturating GAN). Since then, there has been a large amount of work proposing improvements over the original GAN using a variety of tricks [77, 155, 181, 192, 201], network architecture choices [101, 166, 181], objective modifications [4, 13, 54, 74, 97, 105, 138, 140, 142, 152–154, 244], mixtures or ensembles [5, 51, 66, 80, 102, 147, 157, 207, 232], maximum mean discrepancy (MMD) [14, 52, 117, 120, 200], making a connection to reinforcement learning [55, 162], or a combination of these modifications [77, 143, 231]. For a more in-depth discussion of these methods, there are a number of survey papers directed at GAN variants that include a discussion of training challenges and work [79, 86, 135].

Some GANs have been specifically designed with domain adaptation in mind [17, 18, 83, 122, 137, 182, 192, 212, 217]. As a result, the above training stabilization methods can be employed [35, 137, 182, 192, 212]. While these training stability methods could similarly be applied to other adversarial training approaches, they are not typically needed in the non-GAN methods surveyed here.

2.2.2 Evaluation. Once successfully trained, a GAN model can be difficult to evaluate and compare with other models. Multiple approaches and measures have been introduced to evaluate GAN performance. Often researchers have evaluated their models through visual inspection [184] such as performing user studies where participants mark which images they think look more realistic [181]. However, ideally a more automated metric could be found. Past generative models were evaluated by computing log-likelihood [206], but this is not necessarily tractable in GANs [67]. A proxy for log-likelihood is a Parzen window estimate, which was used for early GAN evaluation [69, 133, 153, 206], but in high dimensions (such as images), this could be far from the actual log-likelihood and not even rank models correctly [72, 206]. Thus, there has been much work proposing various evaluation methods for GANs: methods for detecting memorization [13, 49, 69, 133, 166, 206], determining diversity [6, 77, 155, 184], measuring realism [14, 77, 124, 181], and approximating log-likelihood [220]. Xu et al. [224] and Borji [16] survey and compare many of these GAN evaluation methods.

These techniques can be used for evaluating domain adaptation methods used for image translation (a form of image generation but conditioned on an input image) from one domain to another [12, 35, 172, 227, 228, 248]. However, many domain adaptation methods (even those that are adversarial such as those using GANs) are not used for generation but rather for tasks with more easily-defined loss functions, making these techniques largely not needed for adversarial domain adaptation methods. For example, accuracy [12, 18, 19, 35, 57, 61, 83, 122, 210] or AUC scores [165] can be used to evaluate classification, intersection over union or pixel accuracy can be used to

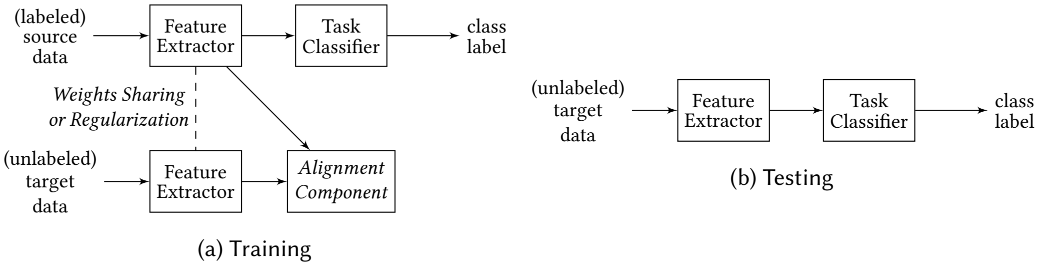


Fig. 3. General network setup for domain adaptation methods learning domain-invariant features. (a) Methods differ in regard to how the domains are aligned during training (the Alignment Component) and whether the feature extractors used on each domain share none, some, or all of the weights between domains. (b) The target data is fed to the domain-invariant feature extractor and then to the task classifier.

evaluate image segmentation [12, 57, 83, 119, 161], and absolute difference can be used to evaluate regression [192].

3 METHODS

In recent years, numerous new unsupervised domain adaptation methods have been proposed, with a growing emphasis on neural network-based approaches. Distinct lines of research have emerged. These include aligning the source domain and target domain distributions, mapping between domains, separating normalization statistics, designing ensemble-based methods, or focusing on making the model target discriminative by moving the decision boundary into regions of lower data density. In addition, others have explored combinations of these approaches. We will describe each of these categories together with recent methods that fall into these categories.

In this survey, we will focus on domain adaptation consisting of one source and one target domain, as is most commonly studied. Another case is multi-source domain adaptation, where there are multiple source domains but still only one target domain. Sun et al. [199] survey multi-source domain adaptation, and since then a number of other methods [24, 75, 82, 134, 160, 242] have been developed for this case. It is also possible to perform multi-target domain adaptation [65], though this case is even more rarely studied.

3.1 Domain-Invariant Feature Learning

Most recent domain adaptation methods align source and target domains by creating a domain-invariant feature representation, typically in the form of a feature extractor neural network. A feature representation is domain-invariant if the features follow the same distribution regardless of whether the input data is from the source or target domain [241]. If a classifier can be trained to perform well on the source data using domain-invariant features, then the classifier may generalize well to the target domain since the features of the target data match those on which the classifier was trained. However, these methods assume that such a feature representation exists and the marginal label distributions do not differ significantly (Section 6).

The general training and testing setup of these methods is illustrated in Figure 3. Methods differ in how they align the domains (the Alignment Component in the figure). Some minimize divergence, some perform reconstruction, and some employ adversarial training. In addition, they differ in weight sharing choices, which will be discussed in Section 4.3. We discuss the various alignment methods below.

3.1.1 Divergence. One method of aligning distributions is through minimizing a divergence that measures the distance between the distributions. Four choices used in various domain adaptation approaches are maximum mean discrepancy, correlation alignment, contrastive domain discrepancy, and the Wasserstein metric.

Maximum mean discrepancy (MMD) [70, 71] is a two-sample statistical test of the hypothesis that two distributions are equal based on observed samples from the two distributions. The test is computed from the difference between the mean values of a smooth function on the two domains' samples. If the means are different, then the samples are likely not from the same distribution. The smooth functions chosen for MMD are unit balls in characteristic reproducing kernel Hilbert spaces (RKHS) since it can be proven that the population MMD is zero if and only if the two distributions are equal [71].

To use MMD for domain adaptation, the alignment component can be another classifier similar to the task classifier. MMD can then be computed and minimized between the outputs of these classifiers' corresponding layers (a slightly different setup than that in Figure 3). Rozantsev et al. [174] use MMD, Long et al. [126] use a multiple kernel variant of MMD (MK-MMD), and later Long et al. [126] develop joint MMD (JMMD) [130]. Bousmalis et al. [19] also tried MMD but found using an adversarial objective performed better in their experiments.

Correlation alignment (CORAL) [197] is similar to MMD with a polynomial kernel, computed from the distance between second-order statistics (covariances) of the source and target features. For domain adaptation, the alignment component consists of computing the CORAL loss between the two feature extractors' outputs (in order to minimize the distance). A variety of distances have been used: Sun et al. [198] use a squared matrix Frobenius norm in Deep CORAL, Zhang et al. [239] use a Euclidean distance in mapped correlation alignment (MCA), others have used log-Euclidean distances in LogCORAL [216] and Log D-CORAL[149], and Morerio et al. [148] use geodesic distances. Zhang et al. [240] generalize correlation alignment to possibly infinite-dimensional covariance matrices in RKHS.

Contrastive domain discrepancy (CCD) [99] is based on MMD but looks at the conditional distributions in order to incorporate label information (unlike CORAL or ordinary MMD). When minimizing CCD, intra-class discrepancy is minimized while inter-class margin is maximized. This has the problem of requiring target labels though, so Kang et al. [99] propose contrastive adaptation networks (CAN) that minimize cross-entropy loss on the labeled target data while alternating between estimating labels for target samples (via clustering) with adapting the feature extractor with the now-computable CCD (using the clusters). This approach outperforms the other methods on the Office dataset as shown in Table 3.

A problem known as "optimal transport" was originally proposed for studying resource allocation such as finding an optimal way to move material from mines to factories [146, 168], but it can also be used to measure the distances between distributions. If the cost of moving each point is a norm (e.g., Euclidean), then the solution to a discrete optimal transport problem can be viewed as a distance: the Wasserstein distance [43] (also known as the earth mover's distance). To align feature and label distributions with this distance, Courty et al. [40] propose joint distribution optimal transport (JDOT). To incorporate this into a neural network, Damodaran et al. [43] propose DeepJDOT.

3.1.2 Reconstruction. Rather than minimizing a divergence, Ghifary et al. [64] and Bousmalis et al. [19] hypothesize that alignment can be accomplished by learning a representation that both classifies the labeled source domain data well and can be used to reconstruct either the target domain data (Ghifary et al.) or both the source and target domain data (Bousmalis et al.). The alignment component in these setups is a reconstruction network – the opposite of the feature extractor network – that takes the feature extractor output and recreates the feature extractor's input (in this case, an image). Ghifary et al. [64] propose deep reconstruction-classification networks (DRCN), using a pair-wise squared reconstruction loss. Bousmalis et al. [19] propose domain separation networks (DSN), using a scale-invariant mean squared error reconstruction loss.

3.1.3 Adversarial. Several varieties of feature-level adversarial domain adaptation methods have been introduced in the literature. In most the alignment component consists of a domain classifier. In one paper this component is instead represented by a network learning an approximate Wasserstein distance, and in another paper the component is a GAN.

A domain classifier is a classifier that outputs whether the feature representation was generated from source or target data. Recall that GANs include a discriminator that tries to accurately predict whether a sample is from the real data distribution or from the generator. In other words, the discriminator differentiates between two distributions, one real and one fake. A discriminator could similarly be designed to differentiate two distributions which instead represent a source distribution and a target distribution, as is done with a domain classifier. Note though that an adversarial domain classifier is used for adaptation, whereas a GAN is used for data generation. The domain classifier is trained to correctly classify the domain (source or target). In this scenario, the feature extractor is trained such that the domain classifier is unable to classify from which domain the feature representation originated. This is a type of zero-sum two-player game [241] as in a GAN (Section 2.2). Typically, these networks are adversarially trained by alternating between these two steps. The feature extractor can be trained to make the domain classifier perform poorly by negating the gradient from the domain classifier with a *gradient reversal layer* [60] when performing back propagation to update the feature extractor weights (e.g., in DANN [1, 60, 61] and VRADA [165]), maximally confusing the domain classifier (when it outputs a uniform distribution over binary labels [209]), or inverting the labels (in ADDA [210]). The domain classifier may also be conditioned on the task classifier predictions when adapting between multimodal distributions [127].

Shen et al. [190] created WDGRL, a modification of DANN, by replacing the domain classifier with a network that learns an approximate Wasserstein distance. This distance is then minimized between source and target domains, which they found to yield an improvement. This method is similar to the divergence methods except here the divergence is learned with a network rather than computed based on statistics (e.g., using mean in MMD or covariance in CORAL). This method outperforms the other methods on the Amazon review dataset as shown in Table 4.

Sankaranarayanan et al. [182] propose Generate to Adapt that uses a GAN as the alignment component. The feature extractor output is both fed to a classifier trained to predict the label (if the input is from the source domain) and also to a GAN trained to generate source-like images (regardless of if the input is source or target). For training stability, they use an AC-GAN [155]. They note one downside of using a GAN for adaptation is that it requires a large training dataset, but a common strategy is to use a pretrained network on a large dataset such as ImageNet. Using this pretraining, even on small datasets (e.g., Office) where the generated images are poor, the network still learns adaptation satisfactorily. Sankaranarayanan et al. [183] similarly develop a similar approach for semantic segmentation.

3.2 Domain Mapping

An alternative to creating a domain-invariant feature representation is mapping from one domain to another. The mapping is typically created adversarially and at the pixel level (i.e., pixel-level adversarial domain adaptation), but not always, as discussed at the end of this section. This mapping can be accomplished with a conditional GAN. The generator performs adaptation at the pixel level by translating a source input image to an image that closely resembles the target distribution. For example, the GAN could change from a synthetic vehicle driving image to one that looks realistic as

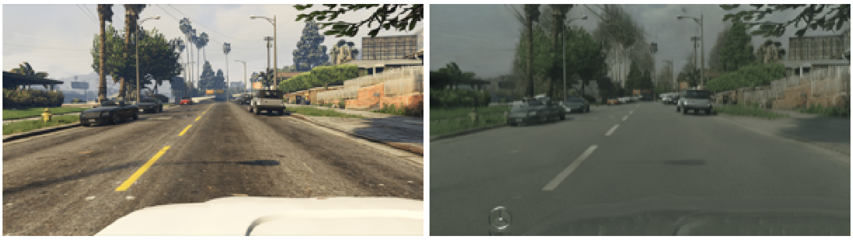


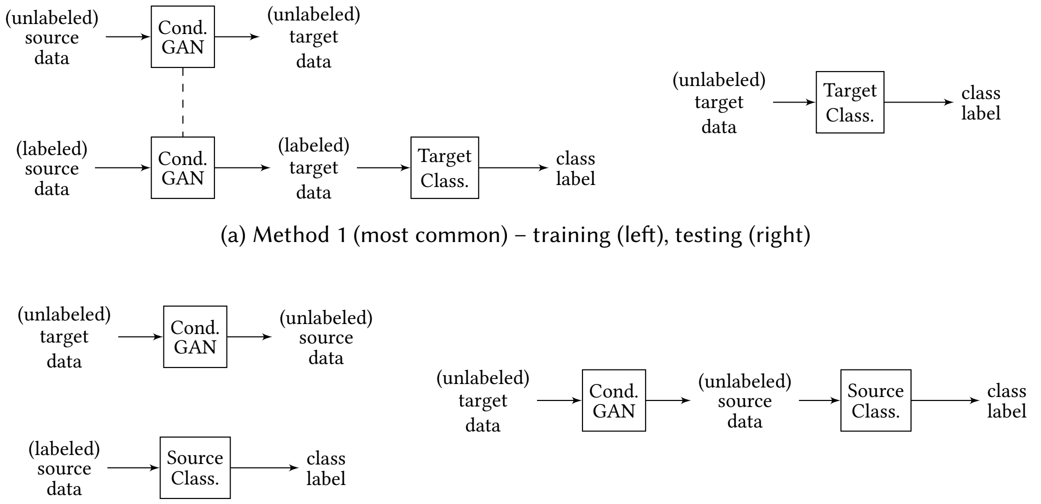
Fig. 4. Synthetic vehicle driving image (left) adapted to look realistic (right) [83].

shown in Figure 4 [35, 83, 172, 228, 248]. A classifier can then be trained on the source data mapped to the target domain using the known source labels [192] or jointly trained with the GAN [18, 83]. We will first discuss how a conditional GAN works followed by the ways it can be employed for domain adaptation.

3.2.1 Conditional GAN for Image-to-Image Translation. The original formulation of a GAN was unconditional, where a GAN only accepted a noise vector as input. Conditional GANs, on the other hand, accept as input other information such as a class label, image, or other data [48, 62, 69, 141]. In the case of image generation, this means that a particular type of image to generate can be specified. One such example is to generate an image of a particular class within an image dataset such as "cat" rather than a random object from the dataset. Another example is conditioning on an input image such as in Figure 4, mapping an input driving image from one domain (synthetic) to an output image in another domain (realistic). Other uses include: transferring style (e.g., make a photo look like a Van Gogh painting) [103, 227, 248], colorizing images [94], generating satellite images from Google Maps data (or vice versa) [94, 227, 248], generating images of clothing from images of people wearing the clothing [228], generating cartoon faces from real faces [172, 202], converting labels to photos (e.g., semantic segmentation output to a photo) [94, 227, 248], learning disentangled representations [31], improving GAN training stability [155], and domain adaptation, which will be discussed in Section 3.2.2.

GANs conditioned on an input image can be used to perform image-to-image translation. These networks can be trained with varying levels of supervision: the dataset may contain corresponding images in the domains (supervised [94, 228]), only a few corresponding images (semi-supervised [59]), or no corresponding images (unsupervised [103, 227, 248]). A popular and general-purpose supervised method is pix2pix, developed by Isola et al. [94]. A commonly used unsupervised method is CycleGAN [248], which is based on pix2pix, or methods similar to CycleGAN including DualGAN [227] and DiscoGAN [103].

Numerous modifications to these approaches have been proposed: one that is multimodal is MUNIT, a multimodal unsupervised image-to-image translator [91]. By assuming a decomposition into style (domain-specific) and content (domain-invariant) codes, MUNIT can generate diverse outputs for a given input image (e.g., multiple possible output images corresponding to the same input image). A modification to CycleGAN explored by Li et al. [118] uses separate batch normalization for each domain (an idea similar to AdaBN discussed in Section 3.3). Mejjati et al. [2] and Chen et al. [32] improve results with attention, learning which areas of the images on which to focus. While CycleGAN and similar approaches use two generators, one for each mapping direction, Benaim et al. [12] developed a method for one-sided mapping that maintains distances between pairs of samples when mapped from the source to the target domain rather than (or in addition to) using a cycle consistency loss, and Fu et al. [57] developed an alternative one-sided mapping using a geometric constraint (e.g., vertical flipping or 90 degree rotation). Royer et al. [172] propose



(a) Method 1 (most common) – training (left), testing (right)

(b) Method 2 – training (left), testing (right)

Fig. 5. Two possible configurations using image-to-image translation for domain adaptation. The conditional GAN and classifier can be trained separately or jointly. Method 1 is the most common. Method 2 is used by one paper. A combination of methods 1 and 2 is used in one paper. The dashed lines between networks indicate that they share weights (or are the same network). Note: this figure does not illustrate the many variants of the conditional GAN component, which often train a generator in each direction (one source to target and one target to source) and use additional losses such as cycle consistency.

XGAN, a dual adversarial autoencoder capable of handling large domain shifts, where possibly an image in the source domain may correspond to multiple images in the target domain or vice versa. They tested mapping human faces to cartoon faces, which was a shift larger than CycleGAN could adequately handle. Choi et al. [35] propose StarGAN, a method for handling multiple domains with a single GAN. Approaches like CycleGAN need a separate generator (or two, one for each direction) for each pair of domains, which is not a scalable solution to many domains. StarGAN, on the other hand, only needs a single generator. This has the added benefit of allowing the generator to learn using all the available data rather than only the data in a specific pair of domains. During training they randomly pick a target domain at each iteration so the generator learns to generate images in all the domains. Anoosheh et al. [3] propose an approach designed for the same purpose as StarGAN but using one generator per domain.

3.2.2 Image-to-Image Translation for Domain Adaptation. While the above approaches map images from one domain to another without the explicit purpose of performing domain adaptation, they can also be used for domain adaptation. For example, the original CycleGAN paper was application agnostic, but others have experimented with applying CycleGAN to domain adaptation [12, 57, 83]. It is important to note though that these image-to-image translation approaches assume that the domain differences are primarily low-level [17, 18, 210].

If unsupervised domain adaptation is performed for classification, adaptation can be accomplished by training an image-to-image translation GAN to map data from source to target, training a classifier on the mapped source images with known labels, and then subsequently testing by feeding unlabeled target through this target-domain classifier [17, 119, 192], as done in SimGAN

[192] and illustrated in Figure 5a. Alternatively, rather than learning a mapping from source to target, the opposite could be done: learn a mapping from target to source, train a classifier on the source images with known labels, and test by feeding target images to the image-to-image translation model (to make them look like source images) followed by the source-domain classifier [28], as illustrated in Figure 5b.

In either of these approaches, if the mapping and the classification models are learned independently, the class assignments may not be preserved. For instance, class 1 may end up being "renamed" to class 2 after the mapping since the mapping was learned ignoring the class labels. This issue can be resolved by incorporating a semantic consistency loss (see Section 4.1) and training the mapping and classification models jointly [19, 83], as done in PixelDA [18].

If there is a way to perform hyperparameter tuning, a third option is possible (combination of Figure 5a and 5b): train a target-domain classifier on the source-to-target GAN (for which the GAN is not used during testing) and a source-domain classifier on the target-to-source GAN (for which the GAN is used during testing). The algorithm may then output a linear combination of the prediction results from the two classifiers [176]. While this approach does improve results, it requires a method of hyperparameter training (see Section 4.7).

All of the above approaches perform pixel-level mapping. An alternative approach is to perform feature-level mapping. Hong et al. [85] use a conditional GAN to learn to make the source features look more like the target features (a distinctly different idea than making the features domain invariant, which was discussed in Section 3.1). They found this particularly helpful for structured domain adaptation (e.g., semantic segmentation, in their case).

Up to this point, these domain mapping methods have used image-to-image translation to map images (or in one case features) from one domain to another and thereby improve domain adaptation performance. Another line of research using pixel-level image generation for domain adaptation is to use a GAN to generate corresponding images in multiple domains and then employ all but the last layer of the discriminator as a feature extractor for a classifier [122, 137]. Liu et al. [122] train a pair of GANs called CoGAN on two domains of images. Mao et al. [137] propose RegCGAN using only one generator and discriminator but including a domain label prepended to the input noise vector.

3.3 Normalization Statistics

Normalization layers such as batch norm [93] are used in most neural networks [185]. These have benefits including allowing for higher learning rates and thus faster training [93], reducing initialization sensitivity [93], smoothing the optimization landscape and making the gradients more Lipschitz [185], and allowing for deeper networks to converge [68, 221]. Each batch norm layer normalizes its input to have zero mean and unit variance. At test time, running averages of the batch norm parameters can be used. Alternatives have been developed including instance norm allowing use in recurrent neural networks [8] and group norm removing the dependence on batch size [221]. However, none of these normalization techniques were developed with domain adaptation in mind. In the case of domain adaptation, the normalization statistics for each domain likely differ. Another line of domain adaptation research involves using per-domain batch normalization statistics.

Li et al. [121] assume that the neural net layer weights learn task knowledge and the batch norm statistics learn domain knowledge. If this is the case, then domain adaptation can be performed by modulating all the batch norm layers' statistics from the source to target domain, a technique they call AdaBN. This has the benefit of being simple, parameter free, and complementary to other adaptation methods.

Carlucci et al. [23] propose AutoDIAL, a generalization of AdaBN. In AdaBN, the target data is not used to learn the network weights but only for adjusting the batch norm statistics. AutoDIAL can utilize the target data for learning the network weights by coupling network parameters between source and target domains. They do this through adding domain alignment layers (DA-layers) that differ for source and target input data before each of the batch norm layers. Generally, batch norm computes a moving average of the statistics on a batch of the layer's input data. However, in AutoDIAL, source and target input data to DA-layers are mixed by a learnable amount before feeding this to batch norm (meaning that the batch norm statistics are now computed over some source and some target data rather than just source data or just target data). This allows the network to automatically learn how much alignment is needed at various points in the network.

3.4 Ensemble Methods

Given a base model such as a neural network or decision tree, an ensemble consisting of multiple models can often outperform a single model by averaging together the models' outputs (e.g., regression) or taking a vote (e.g., classification) [53, 68]. This is because if the models are diverse then each individual model will likely make different mistakes [68]. However, this performance gain corresponds with an increase in computation cost due to the large number of models to evaluate for each ensemble prediction, making ensembles common for some use cases such as competitions but uncommon when comparing models [68]. Despite the incurred cost, several ensemble-based methods have been developed for domain adaptation either using the ensemble predictions to guide learning or using the ensemble to measure prediction confidence for pseudo-labeling target data.

3.4.1 Self-Ensembling. An alternative to using multiple instances of a base model as the ensemble is using only a single model but "evaluating" (via a history or average) the models in the ensemble at multiple points in time during training – a technique called *self-ensembling*. This can be done by averaging over past predictions for each example (by recording previous predictions) [109] or past network weights (by maintaining a running average) [204]. Since an ensemble requires diverse models, these self-ensembling approaches require high stochasticity in the networks, which is provided by extensive data augmentation, varying the augmentation parameters, and including dropout. These methods were originally developed for semi-supervised learning.

French et al. [56] modify and extend these prior self-ensembling methods for unsupervised domain adaptation. They use two networks: a student network and a teacher network. Input images are fed first to stochastic data augmentation (Gaussian noise, translations, horizontal flips, affine transforms, etc.) before being input to both networks. Because the method is stochastic, the augmented images fed to the networks will differ. The student network is trained with gradient descent while the teacher network weights are an exponential moving average (EMA) of the student network's weights. This method outperforms the other methods on the datasets in Table 2. Athiwaratkun et al. [7] show that in at least one experiment stochastic weight averaging [95] can further improve these results.

3.4.2 Pseudo-Labeling. Rather than voting or averaging the outputs of the models in an ensemble, the individual model predictions could be compared to determine the ensemble's confidence in that prediction. The more models in the ensemble that agree, the higher the ensemble's confidence in that prediction. In addition, if performing classification on a particular example, an individual model's confidence can be determined by looking at the last layer's softmax distribution: uniform indicates uncertainty whereas one class's probability much higher than the rest indicates higher confidence. Applying this to domain adaptation, a diverse ensemble trained on source data may be used to label target data. Then, if the ensemble is highly confident, those now-labeled target examples can be used to train a classifier for target data.

This is the method Saito et al. [178] developed called asymmetric tri-training (ATT). Two networks sharing a feature extractor are trained on the labeled source data (i.e., the ensemble in this case is of size two). Those two networks then predict the labels for the unlabeled target data, and if the two agree on the label and have high enough confidence on a particular instance, then the predicted label for that example is assumed to be the true label. After the target data is labeled by the first two networks, the third network (also sharing the same feature extractor) can be trained using the assumed-true labels (pseudo-labels). Diversity in the ensemble is handled with an additional loss (see Section 4.1).

Instead of using an ensemble, Zou et al. [249] rely on just the softmax distribution for the confidence measure. When working with semantic segmentation, they found relying on the prediction confidence for pseudo-labeling results in transferring primarily easy classes while ignoring harder classes. Thus, they additionally propose adding a class-wise weighting term when pseudo-labeling to normalize the class-wise confidence levels and thus balance out the class distribution.

3.5 Target Discriminative Methods

One assumption that has led to successes in semi-supervised learning algorithms is the *cluster assumption* [26]: that data points are distributed in separate clusters and the samples in each cluster have a common label [193]. If this is the case, then decision boundaries should lie in low density regions (i.e., should not pass through regions where there are many data points) [26]. A variety of domain adaptation methods have been explored to move decision boundaries into density regions of lower density. These have typically been trained adversarially.

Shu et al. [193] in virtual adversarial domain adaptation (VADA) and Kumar et al. [108] in co-regularized alignment (Co-DA) both use a combination of variational adversarial training (VAT) developed by Miyato et al. [144] and conditional entropy loss. They are used in combination because VAT without the entropy loss may result in overfitting to the unlabeled data points [108] and the entropy loss without VAT may result in the network not being locally-Lipschitz and thus not resulting in moving the decision boundary away from the data points [193]. Shu et al. [193] additionally propose a decision-boundary iterative refinement step with a teacher (DIRT-T) for use after training to further refine the decision boundaries on the target data, allowing for a slight improvement over VADA. An entropy loss was also used in AutoDIAL [23] but without VAT.

In generative adversarial guided learning (GAGL), Wei et al. [217] propose to let a GAN move decision boundaries into lower-density regions. Using domain alignment methods that learn domain-invariant features like DANN (Section 3.1), typically the data fed to the feature extractor is either source or target data. However, Wei et al. propose to alternate this with feeding generated (fake) images and appending a “fake” label to the task classifier, thus repurposing the task classifier as a GAN discriminator. They found this to have the effect of moving the decision boundaries in the target domain into areas of lower density with a GAN, promoting target-discriminative features as a result.

Saito et al. [179] propose adversarial dropout regularization. Since dropout is stochastic, when they create two instances of the task classifier containing dropout, the resulting networks may produce different predictions. The difference between these predictions can be viewed as a discriminator. Using this discriminator to adversarially train the feature extractor has the effect of producing target discriminative features.

3.6 Combinations

In recent work, researchers have proposed various combinations of the above methods. Domain mapping has been combined with domain-invariant feature learning methods either trained separately (in GraspGAN [17]) or jointly (in CyCADA [83]). Following AdaBN, many researchers started employing domain-specific batch normalization [17, 56, 99, 108, 118]. Kumar et al. [108] propose co-regularized alignment (Co-DA), an approach in which two separate adversarial domain-invariant feature networks are learned with different feature spaces, drawing on ensemble-based methods. Kang et al. [100] combine domain mapping with aligning the models' attention by minimizing an attention-based discrepancy. Deng et al. [47] combine target discriminative methods with self-ensembling. Lee et al. [115] combine target discriminative methods and domain-invariant feature learning with a sliced Wasserstein metric.

Multi-adversarial domain adaptation (MADA) [159] combines adversarial domain-invariant feature learning with ensemble methods for the purpose of better handling multi-modal data. This is accomplished by incorporating a separate discriminator for each class and using the task classifier's softmax probability to weight the loss from each discriminator for unlabeled target samples.

Saito et al. [180] combine elements of adversarial domain-invariant feature learning, ensemble methods, and target discriminative features in their maximum classifier discrepancy (MCD) method. They propose using a shared feature extractor followed by an ensemble (of size two) of task-specific classifiers, where the discrepancy between predictions measures how far outside the support of the source domain the target samples lie. The discriminator in this setup is the combination of the two classifiers. The feature extractor is trained to minimize the discrepancy (i.e., fool the classifiers that the samples are from the source domain) while the classifiers are trained to maximize the discrepancy on the target samples.

4 COMPONENTS

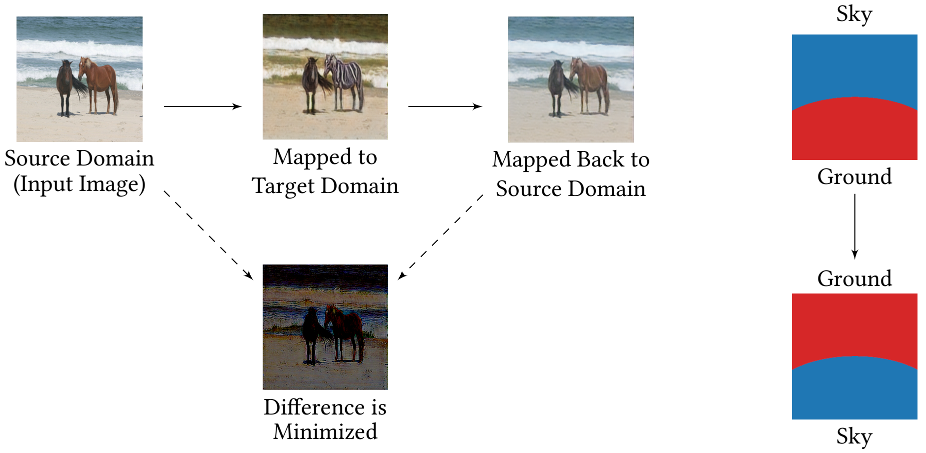
Table 1 summarizes the neural network-based domain adaptation methods we discuss showing components each method uses including what type of adaptation, which loss functions, whether the method uses a generator, and which weights are shared. Below we discuss each of these aspects followed by how the networks are trained, what types of networks can be used, multi-level adaptation techniques, and how to tune the hyperparameters of these methods.

4.1 Losses

4.1.1 Distance. Distance functions play a variety of roles in domain adaptation losses. A distance loss can be used to align two distributions by minimizing a distance function (e.g., MMD) as explained in Section 3.1. If using an ensemble, minimizing a distance function can align the outputs of the ensemble's models: an L1 loss of the difference in predicted target class probabilities from two networks in Co-DA [108] or a squared difference between the predictions of the student and teacher networks in self-ensembling [56]. (Note the squared difference loss is confidence thresholded, i.e., if the max predicted output is below a certain threshold then the squared difference loss is set to zero.)

Some of the described methods have been altered replacing the task loss with one of similarity. Laradji et al. [110] propose M-ADDA, a metric-learning modification to ADDA but with the goal of maximizing the margin between clusters of data points' embeddings. Based on DANN, Pinheiro [163] proposes SimNet, classifying based on how close an embedding is to the embeddings of a random subset of source images for each class. Hsu et al. [88] propose CCN^{++} incorporating a pairwise similarity network (trained with the same class is similar and different classes are dissimilar).

4.1.2 Promote Differences. Methods that rely on multiple networks learning different features (such as to make an ensemble diverse) do so by promoting differences between the networks. Saito et al. [178] train the two classifiers labeling unlabeled data to use different features by adding a norm of the product of the two classifiers' weights. Bousmalis et al. [19] promote different features between two private feature extractors with a soft subspace orthogonality constraint, which is



(a) Cycle Consistency

(b) Semantic Consistency

Fig. 6. (a) Illustration of a cycle-consistency loss using the horses ↔ zebras dataset by Zhu et al. [248]. The difference between the original source image and the reconstructed image (source to target and back to source) is minimized. (b) Example semantic segmentation situation in which the class names are swapped between the input image and the mapped image that would be prevented by including a semantic-consistency loss. The semantic-consistency loss requires that the class assignments are preserved.

similarly used by Liu et al. [123] for text classification. Kumar et al. [108] train the feature extractors to be different by pushing minibatch means apart. Saito et al. [180] maximize the discrepancy between two classifiers using a fixed, shared feature extractor to promote using different features.

4.1.3 Cycle Consistency / Reconstruction. A cycle consistency loss or reconstruction loss is commonly used in domain mapping methods to avoid requiring a dataset of corresponding images to be available in both domains. This is how CycleGAN [248], DualGAN [227], and DiscoGAN [103] can be unsupervised. This means that after translating an image from one domain (e.g., horses) to another (e.g., zebras), the new image can be translated back to reconstruct the original image, as illustrated in Figure 6a. Some variants of this have been proposed such as an L1 loss with a transformation function (e.g., identity, image derivatives, mean of color channels) [192], a feature-level cycle-consistency loss (mapping from source to embedding to target then back to embedding resulting in the same embeddings) [172], or using the loss in one [35] or both directions [83, 172]. Sener et al. [188] enforce cycle consistency in their k-nearest neighbors (k-NN) approach by requiring the distance between any source and target point labeled the same to be less than the distance between any source and target point labeled differently and derive a rule they can solve with stochastic gradient descent.

4.1.4 Semantic Consistency. A semantic consistency loss can be used to preserve class assignments as illustrated in Figure 6b (a segmentation example). The semantic consistency loss requires that a classifier output (or semantic segmentation labeling) from the original source image is the same as the same classifier's output on the pixel-level mapped target output.

4.1.5 Task. Nearly all of the domain adaptation methods include some form of task loss that helps the network learn to perform the desired task. For example, for classification, the goal is to output the ground truth source label, or for semantic segmentation, to label each pixel with the correct ground truth source label. The task loss used is generally a cross-entropy loss, or more specifically the negative log likelihood of a softmax distribution [68] when using a softmax output layer. The exceptions not including a task loss are SimNet [163] that classify based on distance to prototypes of each class, the work by Sener et al. [188] that uses k nearest neighbors, and AdaBN [121] that only adjusts the batch norm layers to the target domain. In addition, the image-to-image translation methods are application agnostic unless trained jointly for domain adaptation.

4.1.6 Adversarial. A variety of methods use a discriminator (or critic) for learning domain-invariant features, realistic image generation, or promoting target discriminative features by forcing a network (either a feature extractor or generator) to produce outputs indistinguishable between two domains (source and target or real and fake). This loss is different than the other losses discussed in this section because this *adversarial loss* is learned [67, 94] (where learning is more than a hyperparameter search) rather than being provided as a predefined function. During training, gradients from the discriminator are used to train the feature extractor or generator (e.g., negated by a gradient reversal layer, Section 3.1.3). This alternates with updating the discriminator itself to make the correct domain classification.

4.1.7 Additions for Specific Problems. Some research focusing on specific problems has resulted in additional losses. For semantic segmentation, Li et al. [119] develop a loss making segmentation boundaries sharper to help when the mapped image-to-image translation images will be used for segmentation, Chen et al. [34] develop a distillation loss in addition to performing location-aware alignment (e.g., "road" is usually at the bottom of each image), Hoffman et al. [84] develop a class-aware constrained multiple instance loss, Zhang et al. [237] develop a curriculum where after learning some high-level properties on easy tasks the segmentation network is forced to follow those properties (interpretations include student-teacher setup or posterior regularization), and Perone et al. [161] apply the self-ensembling method [56] replacing the cross-entropy loss with a consistency loss. For object detection, Chen et al. [33] use two domain classifiers (one on an image-level representation and the other on an instance-level representation) with a consistency regularization between them. For adaptation from synthetic images where it is known which pixels are foreground in the source images, Bousmalis et al. [18] and Bak et al. [9] mask certain losses to only penalize foreground pixel differences. For person re-identification, Wei et al. [218] include a person identity-keeping constraint in their domain mapping GAN.

4.2 Low-Confidence or Low-Relevance Rejection

Given a measure of confidence, performance may increase if we can reject data points for training the target classifier that are not of sufficient confidence. This, of course, assumes our confidence measurement is accurate enough. Saito et al. [178] used the label agreement of an ensemble combined with the softmax distribution output (uniform is not confident, one probability much higher than the rest is confident). Sener et al. [188] used the label agreement of the k nearest source data points. If the confidence is to low, then the example is rejected and not used in training until if later on when re-evaluated it is determined to be sufficiently confident. Inoue et al. [92] used an object detector's prediction probability as a measure of confidence, only using high-confidence detections for fine-tuning an object detection network. Similarly, a rejection approach could be used if we have a measure of relevance. For text classification, Zhang et al. [236] weight examples by their relevance to their target aspect based on a small set of positive and negative keywords (a form of weak supervision).

4.3 Weight Sharing

Methods employ different amounts of sharing network weights between domains or regularizing the weights to be similar. Most methods completely share weights between the feature extractors used on the source and target domains (as shown in Table 1). However, some techniques do not. Since deep networks consist of many layers, allowing them to represent hierarchical features, Long et al. [126] propose copying the lower layers from a network trained on the source domain and adapting higher layers to the target domain with MK-MMD since higher layers do not transfer well between domains. In CoGAN, Liu et al. [122] share the first few layers of the generators and the last few layers of the discriminators, making the assumption that the domains share high-level representations. In AdaBN, Li et al. [121] assume domain knowledge is stored in the batch norm statistics, so they share all weights except for the batch norm statistics. French et al. [56] define the teacher network as an exponential moving average of the student network's weights (a type of ensemble). Instead of sharing weights, Rozantsev et al. [173, 174] propose two variants: regularizing weights to be similar but not penalizing linear transformations and transforming the weights from the source network to the target network with small residual networks. Bousmalis et al. [19] propose domain separation networks (DSN): learning source-specific, target-specific, and shared features where the "shared" source domain encoder and "shared" target domain encoder do share weights, but the "private" source domain encoder and "private" target domain encoders do not. Others have similarly explored this idea of shared vs. specific features [22, 123, 169].

4.4 Training Stages

Some have trained networks for domain adaptation in stages. Tzeng et al. [210] train a source classifier first followed by adaptation. Taigman et al. [202] use a pre-trained encoder during adaptation. Bousmalis et al. [17] in GraspGAN first train the domain-mapping network followed by the domain-adversarial network. Hoffman et al. [83] in CyCADA train their many components in stages because it would not all fit into GPU memory at once.

Other methods train the domain adaptation networks jointly, which using an adversarial approach is done by alternating between training the discriminator and the rest of the networks (Sections 2.2 and 3.1.3). However, variations exist for some other methods. Saito et al. [178] in ATT cycle through generating training the source networks, generating pseudo-labels, and training the target network. Zou et al. [249] alternate between pseudo-labeling the target data and re-training the model using the labels (a form of self-training). Wei et al. [217] in GAGL alternate between feeding in real source and target data and the fake images generated by a GAN. Sener et al. [188] alternate between k-nearest neighbors and performing gradient descent.

4.5 Multi-Level

Some adaptation methods perform adaptation at more than one level. As discussed in Section 3.6, GraspGAN [17] and CyCADA [83] perform pixel-level adaptation with domain mapping and feature-level adaptation with domain-invariant feature learning. Hoffman et al. [83] found that performing both levels of adaptation significantly improves accuracy: using domain mapping to capture low-level image domain shifts and learning domain-invariant features to handle larger domain shifts than what pure domain mapping methods can support. Following this idea, Tsai et al. [208] make semantic segmentation predictions and perform domain-invariant feature learning at multiple levels in their semantic segmentation network, and Zhang et al. [235] perform domain-invariant feature learning at multiple levels while automatically learning how much to align to each level. Chen et al. [33] perform domain-invariant feature learning at both image and instance levels for object detection but also include a consistency regularization between the two domain classifiers.

4.6 Types of Networks

Nearly all of the surveyed approaches focus on learning from image data and use convolutional neural networks (CNNs) such as ResNet-50 or Inception (Table 3). Wang et al. [214] explore the use of attention networks and Kang et al. [100] a combination of CNNs and attention. In the case of time-series data, Purushotham et al. [165] propose instead using a variational recurrent neural network (RNN) [37] or LSTM (a type of RNN) [81] rather than a CNN. The RNN learns the temporal relationships while adversarial training is used to achieve domain adaptation. For text classification (a type of natural language processing), Liu et al. [123] also use LSTMs while Zhang et al. [236] found a CNN to work just as well as RNNs or bi-LSTMs in their experiments. For relation extraction (another type of natural language processing), Fu et al. [58] also use a CNN. For time-series speech recognition, Zhao et al. [243] use bi-LSTMs while Hosseini-Asl et al. [87] used a combination of CNNs and RNNs. In the related problem of domain generalization, a combination of CNNs and RNNs have been used for handling a radio spectrogram changing through time to identify sleep stages [245].

4.7 Hyperparameter Tuning

Normal supervised learning-based hyperparamenter tuning methods do not carry over to unsupervised domain adaptation [19, 61, 128, 129, 148, 161, 212]. A common supervised learning approach is to split the training data into a smaller training set and a validation set. After repeatedly altering the hyperparameters, retraining the model, and testing on this validation set for each set of hyperparameters, the model yielding the highest validation set accuracy is selected. Another option is cross validation. However, in unsupervised domain adaptation, there are now two domains, and the data for the target domain may not include any labels. When evaluating domain adaptation approaches on common datasets, generally the target data does contain labels, so work by some groups [19, 23, 108, 176, 193, 212, 217] do use some labeled target data (or all of it [128, 190]) for hyperparameter tuning, which can be interpreted as an upper bound on how well the method could perform [212]. For example, some [23, 129] tuned for Office on one W labeled example per class on the A →W task, while others [176, 217] tuned with a validation set of 1000 randomly sampled target examples. Using any labeled target data is not ideal because real-world testing will not include labels for tuning (unless it is semi-supervised, in which case semi-supervised learning is recommended in Section 6).

One tuning method not requiring labeled target data is *reverse validation* [61], which is a variant of *reverse cross validation* [247]. For a set of hyperparameters, the *reverse validation risk* can be estimated by first splitting source (labeled) and target (unlabeled) data into training and validation sets. Then, the labeled source and unlabeled target data is used to learn a classifier (as is normally done). Next, this forward classifier is used to label the target data and a new reverse classifier is learned (with the same algorithm) using the pseudo-labeled target data (as “source”) and unlabeled source data (as “target”, i.e., ignoring the known labels). This reverse classifier is evaluated on the source validation data to measure the reverse validation risk. Ganin et al. [61] found this method works better if the reverse classifier is initialized with the weights of the forward classifier and if using early stopping on the source validation set and a pseudo-labeled target validation set. Finally, hyperparameters are selected (e.g., grid search, random search, Bayesian optimization, or other gradient-free optimization methods such as those implemented in Nevergrad [167]) that minimize this reverse validation risk.

Table 1. Comparison of different neural network based domain adaptation methods based on method of adaptation (domain-invariant feature learning [DI], domain mapping [DM], normalization [N], ensemble [En], target discriminative [TD]), various loss functions (distance, promoting different features, cycle consistency, semantic consistency, task, feature- or pixel-level adversarial), usage of a generator, and which weights are shared (in the feature extractor).

Name	Method	Loss Functions					Adversarial Loss		Generator	Shared Weights
		Distance	Diff.	Cycle	Sem.	Task	Feature	Pixel		
CAN[99]	DI,N	CCD				✓				not BN
French et al.[56]	En,N	sq. diff.				✓				EMA
Co-DA[108][a]	DI,En,N,TD	L1	✓			✓	✓			optional
VADA[193][a]	DI,TD					✓	✓			✓
DeepJDOT[43]	DI	JDOT				✓				✓
CyCADA[83]	DI,DM			✓	✓	✓	✓	✓	✓	
Gen. to Adapt[182]	DI					✓	✓		✓	✓
SimNet[163]	DI	prototypes					✓			
MADA[159]	DI,En					✓	✓			✓
MCD[180]	DI,En,TD	✓	✓			✓	✓			
GAGL[217]	DI,TD					✓	✓	✓	✓	✓
SBADA-GAN[176][b]	DM				✓	✓		✓	✓	
MCA[239]	DI	MCA				✓				✓
CCN^{++}[88]	DI	clusters					✓			✓
M-ADDA[110]	DI	clusters				✓	✓			
Rozant. et al.[174]	DI	MMD				✓				regularize
XGAN[172]	DM			✓			✓	✓	✓	some
StarGAN[35]	DM			✓				✓	✓	✓
PixelDA[18]	DM				✓	✓		✓	✓	✓
AutoDIAL[23]	N,TD					✓				not BN
AdaBN[124]	N									not BN
JAN-A[130]	DI	JMMD				✓	✓			✓
LogCORAL[216]	DI	logCOR, mean				✓				✓
Log D-CORAL[149]	DI	logDCOR				✓				✓
VRADA[165]	DI					✓	✓			✓
ATT[178]	En		✓			✓				✓
SimGAN[192]	DM							✓	✓	N/A[c]
ADDA[210]	DI					✓	✓			
CycleGAN[248]	DM			✓				✓	✓	[d]
RegCGAN[137]	DM					✓	✓		✓	✓
Sener et al.[188]	DI	k-NN								✓
DSN[19]	DI		✓	✓		✓	✓			some
DRCN[64]	DI			✓		✓				✓
CoGAN[122]	DM					✓	✓		✓	some
Deep CORAL[198]	DI	CORAL				✓				✓
DANN[1, 60, 61]	DI					✓	✓			✓
DAN[126]	DI	MK-MMD				✓				low
Tzeng et al.[209][e]	DI					✓	✓			✓

[a] also incorporate virtual adversarial training [144]

[b] also a self-labeled classification loss (learn label on source images, pseudo-label mapped target to source)

[c] maps to target domain so only have feature extractor for target (part of the classifier)

[d] unspecified; originally not applied to domain adaptation, but later used for this [12, 57, 83]

[e] semi-supervised for some classes, i.e., requires some labeled target data for some of the classes

Alternatively, given some domain knowledge, one may devise relevant measures of similarity between the domains and tune parameters to increase the similarity. For example, French et al. [56] were able to improve performance on the challenging problem of MNIST → SVHN by tuning data augmentation hyperparameters for MNIST to match pixel intensities apparent in the SVHN dataset. By doing this, they were able to improve the state-of-the-art to 97.0% (Table 2).

5 RESULTS

Tables 2 through 5 summarize the results of evaluating many of these methods on datasets used for image classification as well as sentiment analysis. Care must be taken in the extent to which conclusions are drawn from comparing published numbers in different papers since the provided accuracies are for different network architectures, hyperparameters, amount of data augmentation, random initializations (or averages over a number of them), etc. and the methods may perform differently in other application areas. However, interestingly, at least one method in each of the categories of surveyed gives promising results on at least one of the datasets.

With domain-invariant feature learning with the contrastive domain discrepancy, CAN [99] has the highest performance on the Office dataset (Table 3). By using adversarial domain-invariant feature learning, WDGRL generally outperforms the other methods on the Amazon review dataset (Table 4) and Generate to Adapt is second highest of the methods evaluated on the Office dataset. By using adversarial pixel-level domain mapping, SBADA-GAN [176] obtains the highest accuracy on MNIST→MNIST-M (Table 2). AutoDIAL [23], a normalization statistics method, does on-par with CAN and Generate to Adapt in two of Office adaptation tasks. The self-ensembling method by French et al. [56] outperforms all other methods on the datasets in Table 2, and Co-DA [108] comes close using an ensemble (of size two) of adversarial domain-invariant feature networks. CyCADA increases accuracy from 54% to 82% for a synthetic season adaptation dataset [83] by combining both adversarial domain-invariant feature learning and domain mapping.

A number of these promising methods use adversarial techniques, which may be a key ingredient in solving domain adaptation problems. Adversarial approaches may be helpful on certain datasets (e.g., WDGRL on the Amazon review dataset on Office), certain types of data (e.g., VRADA was developed for time series data rather than image data), or may not require as extensive of tuning (e.g., Co-DA on MNIST→SVHN). Or adversarial training may be an additional tool to incorporate into existing non-adversarial methods. For instance, promising non-adversarial methods such as AutoDIAL and by French et al. could be combined with adversarial methods (see Section 8.3). In fact, Long et al. [130] develop both JAN and then the adversarial version JAN-A, and JAN-A on average outperformed JAN on the Office dataset. CAN [99], which presently is the highest on the Office dataset, might also be improved by incorporating an adversarial component to it as in Long et al. [130].

Interestingly, French et al. by far outperform all other methods on MNIST→SVHN, though this requires a problem-specific data augmentation and hyperparameter tuning. This may indicate that for some problems, maybe in particular the more challenging domain adaptation problems, hyperparameter tuning for a specific dataset may be of utmost importance. Possibly if other domain adaptation methods similarly were tuned appropriately, they would also experience large improvements. This is an area of research requiring further work (see Section 8.2). However, Co-DA [108] is not far behind on SVHN→MNIST and MNIST→MNIST-M and is the closest on MNIST→SVHN, achieving 81.7% compared with 97.0%. A great advantage of Co-DA is that it does not require highly-problem-specific tuning on MNIST→SVHN as required by French et al. (without they only achieved 37.5%). Possibly some components of Co-DA such as the adversarial domain adaptation or virtual adversarial training may be partially responsible for the decrease in hyperparameter sensitivity.

Table 2. Classification accuracy (source → target, mean ± std %) of different neural network based domain adaptation methods on various computer vision datasets (only including those used in > 2 papers). Adversarial approaches denoted by *.

Name	MNIST and USPS		MNIST and SVHN		MNIST[-M]	Synthetic to Real	
	MN → US	US → MN	SV → MN	MN → SV	MN → MN-M	SYN$_N$ → SV	SYN$_S$ → GTSRB
Target only (i.e., if we had the target labels)	96.3 ± 0.1 [83] 96.5 [18]	99.2 ± 0.1 [83]	99.2 ± 0.1 [83] 99.5 [19] 99.51 [60]		96.4 [18] 98.7 [19] 98.91 [60]	92.44 [60] 92.4 [19]	99.87 [60] 99.8 [19]
French et al.[56]	98.2	99.5	99.3	37.5 97.0[a]		97.1	99.4
Co-DA[108][b]*			98.6	81.7	97.5	96.0	
DIRT-T[193][b]*			99.4	76.5	98.7	96.2	99.6
VADA[193][b]*			94.5	73.3	95.7	94.9	99.2
DeepJDOT[43]	95.7	96.4	96.7		92.4		
CyCADA[83]*	95.6 ± 0.2	96.5 ± 0.1	90.4 ± 0.4				
Gen. to Adapt[182]*	92.8 ± 0.9	90.8 ± 1.3	92.4 ± 0.9				
SimNet[163]*	96.4	95.6			90.5		
MCD[180]*	96.5 ± 0.3	94.1 ± 0.3	96.2 ± 0.4				94.4 ± 0.3
GAGL[217][b]*			96.7	74.6	94.9	93.1	97.6
SBADA-GAN[176][b]*	97.6	95.0	76.1	61.1	99.4		96.7
MCA[239]			96.6		96.8	89.0	
CCN^{++}[88]*			89.1				
M-ADDA[110]*	98	97					
Rozantsev et al.[174]	60.7	67.3					
PixelDA[18]*	95.9				98.2		
ATT[178]			85.0	52.8	94.0	92.9	96.2
ADDA[210]*	89.4 ± 0.2	90.1 ± 0.8	76.0 ± 1.8				
RegCGAN[137]*	93.1 ± 0.7	89.5 ± 0.9					
DTN[202]*			84.4				
Sener et al.[188]			78.8	40.3	86.7		
DSN[19][b]*	91.3 [18]		82.7		83.2	91.2	93.1
DRCN[64]	91.80 ± 0.09	73.67 ± 0.04	81.97 ± 0.16	40.05 ± 0.07			
CoGAN[122]*	91.2 ± 0.8	89.1 ± 0.8			62.0 [18]		
DANN[60, 61]*	85.1 [18]		71.07 70.7 [19] 71.1 [178] 73.6 [83]	35.7 [178]	81.49 77.4 [19] 81.5 [178]	90.48 90.3 [19, 178]	88.66 88.7 [178] 92.9 [19]
DAN[126]	81.1 [18]		71.1 [19]		76.9 [19]	88.0 [19]	91.1 [19]
Source only (i.e., no adaptation)	78.9 [18] 82.2 ± 0.8 [83]	69.6 ± 3.8 [83]	59.19 [60] 59.2 [19] 67.1 ± 0.6 [83]		56.6 [19] 57.49 [60] 63.6 [18]	86.65 [60] 86.7 [19]	74.00 [60] 85.1 [19]

[a] problem-specific hyperparameter tuning of data augmentation to match pixel intensities of target domain images
[b] hyperparameter tuned on some labeled target data

Table 3. Classification accuracy (source → target, mean ± std %) of different neural network based domain adaptation methods on the Office computer vision dataset. Adversarial approaches denoted by *.

Name	Office (Amazon, DSLR, Webcam)					
	A → W	D → W	W → D	A → D	D → A	W → A
CAN[99][a]	94.5 ± 0.3	99.1 ± 0.2	99.8 ± 0.2	95.0 ± 0.3	78.0 ± 0.3	77.0 ± 0.3
Gen. to Adapt[182][a]*	89.5 ± 0.5	97.9 ± 0.3	99.8 ± 0.4	87.7 ± 0.5	72.8 ± 0.3	71.4 ± 0.4
SimNet[163][a]*	88.6 ± 0.5	98.2 ± 0.2	99.7 ± 0.2	85.3 ± 0.3	73.4 ± 0.8	71.8 ± 0.6
MADA[159][a]*	90.0 ± 0.1	97.4 ± 0.1	99.6 ± 0.1	87.8 ± 0.2	70.3 ± 0.3	66.4 ± 0.3
AutoDIAL[23][bc]	84.2	97.9	99.9	82.3	64.6	64.2
CCN^{++}[88][d]*	78.2	97.4	98.6	73.5	62.8	60.6
Rozantsev et al.[174]	76.0	96.7	99.6			
AdaBN[124][b]	74.2	95.7	99.8	73.1	59.8	57.4
JAN-A[130][a]*	86.0 ± 0.4	96.7 ± 0.3	99.7 ± 0.1	85.1 ± 0.4	69.2 ± 0.4	70.7 ± 0.5
LogCORAL[216]	70.2 ± 0.6	95.5 ± 0.1	99.5 ± 0.3	69.4 ± 0.5	51.2 ± 0.3	51.6 ± 0.5
Log D-CORAL[149]	68.5	95.3	98.7	62.0	40.6	40.6
ADDA[210][a]*	75.1	97.0	99.6			
Sener et al.[188]	81.1	96.4	99.2	84.1	58.3	63.8
DRCN[64]	68.7 ± 0.3	96.4 ± 0.3	99.0 ± 0.2	66.8 ± 0.5	56.0 ± 0.5	54.9 ± 0.5
Deep CORAL[198]	66.4 ± 0.4	95.7 ± 0.3	99.2 ± 0.1	66.8 ± 0.6	52.8 ± 0.2	51.5 ± 0.3
DANN[60, 61]*	67.3 ± 1.7 72.6 ± 0.3 [64] 73.0 [174, 210]	94.0 ± 0.8 96.4 ± 0.1 [64] 96.4 [174, 210]	93.7 ± 1.0 99.2 ± 0.3 [64] 99.2 [174, 210]	67.1 ± 0.3 [64]	54.5 ± 0.4 [64]	52.7 ± 0.2 [64]
DAN[126]	68.5 ± 0.4 63.8 ± 0.4 [198] 64.5 [174] 68.5 [210]	96.0 ± 0.3 94.6 ± 0.5 [198] 95.2 [174] 96.0 [210]	99.0 ± 0.2 98.6 [174] 98.8 ± 0.6 [198] 99.0 [210]	67.0 ± 0.4 65.8 ± 0.4 [198]	54.0 ± 0.4 52.8 ± 0.4 [198]	53.1 ± 0.3 51.9 ± 0.5 [198]
Tzeng et al.[209] [e]*	59.3 ± 0.6	90.0 ± 0.2	97.5 ± 0.1	68.0 ± 0.5	43.1 ± 0.2	40.5 ± 0.2
Source only (i.e., no adaptation)	62.6 [210][a]	96.1 [210][a]	98.6 [210][a]			

[a] with ResNet-50 network

[b] with Inception-based network

[c] hyperparameter tuned on one W labeled example per class on A →W task (see [129])

[d] with ResNet-18 network

[e] semi-supervised for some classes, but evaluated on 16 hold-out categories for which the labels were not seen during training

6 THEORY

Having surveyed domain adaptation methods, we now address the question of when adaptation may be beneficial. Ben-David et al. [11] develop a theory answering this in terms of an ideal predictor on both domains, Zhao et al. [241] further this theory by removing the dependence on a joint ideal predictor while focusing on domain-invariant feature learning methods, and Le et al. [112] develop theory looking beyond domain-invariant methods. These theoretical results can help answer two questions: (1) when will a classifier (or other predictor) trained on the source data perform well on the target data, and (2) given a small number of labeled target examples, how can they best be used during training to minimize target test error?

Table 4. Classification accuracy comparison for domain adaptation methods for sentiment analysis (positive or negative review) on the Amazon review dataset [15][a] with domains books (B), DVD (D), electronics (E), and kitchen (K). Adversarial approaches denoted by *.

Source → Target	**DANN**[61][b]*	**DANN**[61][c]*	**CORAL**[197][d]	**ATT**[178][c]	**WDGRL**[190][ce]*	**No Adapt.**[197][f]
B → D	82.9	78.4		80.7	83.1	
B → E	80.4	73.3	76.3	79.8	83.3	74.7
B → K	84.3	77.9		82.5	85.5	
D → B	82.5	72.3	78.3	73.2	80.7	76.9
D → E	80.9	75.4		77.0	83.6	
D → K	84.9	78.3		82.5	86.2	
E → B	77.4	71.3		73.2	77.2	
E → D	78.1	73.8		72.9	78.3	
E → K	88.1	85.4	83.6	86.9	88.2	82.8
K → B	71.8	70.9		72.5	77.2	
K → D	78.9	74.0	73.9	74.9	79.9	72.2
K → E	85.6	84.3		84.6	86.3	

[a] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[b] using 30,000-dimensional feature vectors from marginalized stacked denoising autoencoders (mSDA) by Chen et al. [30], which is an unsupervised method of learning a feature representation from the training data

[c] using 5000-dimensional unigram and bigram feature vectors

[d] using bag-of-words feature vectors including only the top 400 words, but suggest using deep text features in future work

[e] the best results on target data for various hyperparameters

[f] using bag-of-words feature vectors

Answering the first question, labeled source data and unlabeled target data are both required (unsupervised). Answering the second question, additionally some labeled target data are required (semi-supervised). We will first review the theoretical bounds followed by a discussion of what insights these bounds provide into answering the above two questions. Ben-David et al. [11] also address the case of multiple source domains, as do Mansour et al. [136]. In this paper, we have focused on the cases containing only one source and one target (as is common in the methods we survey).

6.1 Unsupervised

6.1.1 Shared Hypothesis Space. Ben-David et al. [11] propose setting a bound on the target error based on the source error and the divergence between the source and target domains. The empirical source error is easy to obtain by first training and then testing a classifier. However, the divergence between the domains cannot be directly obtained with standard methods like Kullback-Leibler divergence due to only having a finite number of samples from the domains and not assuming any particular distribution. Thus, an alternative is to measure it using a classifier-induced divergence called $\mathcal{H}\Delta\mathcal{H}$-divergence. Estimates of this divergence with finite samples converges to the real $\mathcal{H}\Delta\mathcal{H}$-divergence. This divergence can be estimated by measuring the error when getting a classifier to discriminate between the unlabeled source and target examples; though, it is often intractable to find the theoretically-required divergence upper bound. Using the empirical source error $\hat{\epsilon}_S(h)$, the $\mathcal{H}\Delta\mathcal{H}$-divergence between source and target samples $d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_S, \hat{\mathcal{D}}_T)$, and ideal predictor error λ^* using the optimal hypothesis for the source and target, the target error $\epsilon_T(h)$ can be bounded as shown in Equation 2 (using the form given by Zhao et al. [241]), $\forall h \in \mathcal{H}$ with probability at least

Table 5. List and description of computer vision datasets from Tables 2 and 3

Computer Vision Datasets used for Domain Adaptation	
MNIST[113][a]	This is a binary (mostly black and white, but actually grayscale due to anti-aliasing) handwritten digit dataset (digits 0-9), which stands for "modified NIST." It is based on the National Institute of Standards and Technology's (NIST) Special Database 1 and 3, one of which was easier than the other, so MNIST is a combination of the two that are size normalized to fit in a 20x20 box preserving the aspect ratio and centered in a 28x28 pixel image.
MNIST-M[61][b]	This is a modification of MNIST where the digits are blended with random patches from BSDS500 dataset color photos.
USPS[114][c]	This is another handwritten digit dataset (digits 0-9). It consists of handwritten zipcodes scanned and segmented by the U.S. Postal Service (USPS). They were size normalized to 16x16 pixels preserving the aspect ratio. The values are normalized to be between -1 and 1.
SVHN[151][d]	The Streetview House Numbers (SVHN) consists of single digits extracted from images of urban house numbers in Google Street View. The digits have been size normalized to 32x32 pixels.
SYN$_{\mathbf{N}}$[61][b]	Ganin et al. [61] used Microsoft Windows fonts to create a synthetic digit dataset ("Syn Numbers") consisting of 1-3 digit numbers with various positions, orientation, background color, stroke color, and amount of blur.
SYN$_{\mathbf{S}}$[145][e]	This is a synthetic sign dataset created from modifications to Wikipedia pictograms of traffic signs. It consists of 100,000 images and 43 classes of signs.
GTSRB[196][f]	The German Traffic Signs Recognition Benchmark (GTSRB) is a dataset created from video taken driving around Germany. It consists of about 50,000 images and 43 classes of signs.
Office[177][g]	This dataset consists of 31 classes of objects in three different domains: Amazon (taken from its online website; medium resolution and studio lighting), DSLR (taken with a digital SLR camera; high resolution and in a real-world environment), and Webcam (taken with a 640x480 computer webcam; have noise, artifacts, and white balance issues). Note: due to Office's small size, some networks [61, 174, 198] were pre-trained on ImageNet.

[a] http://yann.lecun.com/exdb/mnist/

[b] See Ganin's website http://yaroslav.ganin.net/ for links to download.

[c] This can be found on various sites and some Github repositories. One such place: https://web.stanford.edu/~hastie/ElemStatLearn/data.html

[d] http://ufldl.stanford.edu/housenumbers

[e] The synthetic dataset linked to on: http://graphics.cs.msu.ru/en/research/projects/imagerecognition/trafficsign

[f] http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset

[g] http://ai.bu.edu/adaptation.html

$1 - \delta$ for $\delta \in (0, 1)$.

$$\epsilon_T(h) \leq \hat{\epsilon}_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_S, \hat{\mathcal{D}}_T) + \lambda^* + O\left(\sqrt{\frac{d \log n + \log(\frac{1}{\delta})}{n}}\right) \tag{2}$$

Zhao et al. [241] develop another upper bound that removes the reliance on λ^*. Let $\mathcal{H} \subseteq [0,1]^{\mathcal{X}}$, $\tilde{\mathcal{H}} := \{\text{sgn}(|h(x) - h'(x)| - t) \,|\, h, h' \in \mathcal{H}, 0 \leq t \leq 1\}$, $\langle \mathcal{D}_S, f_S \rangle$ and $\langle \mathcal{D}_T, f_T \rangle$ be the source and target domains (the true distributions, not empirical). The target error can then be bounded by the source error $\epsilon_S(h)$, the discrepancy between marginal distributions $d_{\tilde{\mathcal{H}}}(\mathcal{D}_S, \mathcal{D}_T)$, and the distance between the optimal source and target labeling functions $\forall h \in \mathcal{H}$, as shown in Equation 3.

$$\epsilon_T(h) \leq \epsilon_S(h) + d_{\tilde{\mathcal{H}}}(\mathcal{D}_S, \mathcal{D}_T) + \min\{\mathbb{E}_{\mathcal{D}_S}[|f_S - f_T|], \mathbb{E}_{\mathcal{D}_T}[|f_S - f_T|]\} \tag{3}$$

Zhao et al. [241] also develop an information-theoretic lower bound for target error. Let the labeling function $Y = f(X) \in \{0,1\}$, the prediction function $\hat{Y} = h(g(X)) \in \{0,1\}$, and Z be the intermediate representation output by a shared feature extractor used on source and target domain data. If the Jensen-Shannon distance $d_{JS}(\mathcal{D}_S^Y, \mathcal{D}_T^Y) \geq d_{JS}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$ and the Markov chain $X \xrightarrow{g} Z \xrightarrow{h} \hat{Y}$ holds, then Equation 4 provides a lower bound on the source and target error.

$$\epsilon_S(h \circ g) + \epsilon_T(h \circ g) \geq \frac{1}{2}\left(d_{JS}(\mathcal{D}_S^Y, \mathcal{D}_T^Y) - d_{JS}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)\right)^2 \tag{4}$$

6.1.2 *Different Hypothesis Spaces.*

Le et al. [112] develop an upper bound that allows for different hypothesis spaces for source and target functions, possibly non-deterministic labeling, and any bounded or continuous loss. If l is a bounded or continuous loss, $x \sim \mathbb{P}^s$ (source) and $x \sim \mathbb{P}^t$ (target), $T : \mathcal{X}^s \rightarrow \mathcal{X}^t$ and $K := T^{-1}$ (bijective mapping), $R(\theta) = \mathbb{E}_{p(x,y)}[l(y, h_\theta(x))]$ for θ parameterizing a hypothesis set $\mathcal{H} = \{h_\theta | \theta \in \Theta\}$, $\Delta R(h^s, h^t) := |R^t(h^t) - R^s(h^s)|$, $y \in \{-1, 1\}$, M is the number of labels, $\mathbb{P}^\# := K_\# \mathbb{P}^t$ is the pushforward probability distribution transporting $\mathbb{P}^t$ via K, $\Delta p(y|x) := p^t(y|T(x)) - p^s(y|x)$ for the true source and target labeling functions $p^s(y|x)$ and $p^t(y|x)$, where $WS_c(\mathbb{P}^s, \mathbb{P}^\#)$ denotes the Wasserstein-1 distance between the source and target distributions with a cost function $c(x, x') = 1_{x \neq x'}$ (1 if $x \neq x'$, otherwise 0), then Equation 5 provides an upper bound for the variance between a general loss on the source and target predictions.

$$\Delta R(h^s, h^t) \leq M\left(WS_c(\mathbb{P}^s, \mathbb{P}^\#) + \min\{\mathbb{E}_{\mathbb{P}^\#}[\|\Delta p(y|x)\|_1], \mathbb{E}_{\mathbb{P}^s}[\|\Delta p(y|x)\|_1]\}\right) \tag{5}$$

6.2 Semi-Supervised

In the semi-supervised case, a linear combination of the source and target errors is computed [11], called the α-error. A bound can be calculated on the true α-error based on the empirical α-error. Finding the minimum α-error depends on the empirical α-error, the divergence between source and target, and the number of labeled source and target examples. Experimentation can be used to empirically determine the values of α that will perform well. Ben-David et al. [11] also demonstrate the process on sentiment classification, illustrating that the optimum uses non-trivial values.

The bound is given in Equation 6. If S is a labeled sample of size m with $(1 - \beta)m$ points drawn from the source distribution and βm from the target distribution, then with at least probability $1 - \delta$ for $\delta \in (0, 1)$:

$$\begin{aligned} \epsilon_T(\hat{h}) \leq & \epsilon_T(h_T^*) + 4\sqrt{\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta}}\sqrt{\frac{2d\log(2(m+1)) + 2\log(\frac{8}{\delta})}{m}} + \\ & 2(1-\alpha)\left(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{8}{\delta})}{m'}} + \lambda\right) \end{aligned} \tag{6}$$

Here, $\hat{h} \in \mathcal{H}$ is the empirical minimizer of the α-error on S given by $\hat{\epsilon}_\alpha(h) = \alpha\hat{\epsilon}_T(h) + (1-\alpha)\hat{\epsilon}_S(h)$ and $h_T^* = \min_{h \in \mathcal{H}} \epsilon_T(h)$ is the target error minimizer.

The optimum α is then:

$$\alpha^*(m_T, m_S; D) = \begin{cases} 1 & m_T \geq D^2 \\ \min\{1, \nu\} & m_T \leq D^2 \end{cases} \tag{7}$$

Here, $m_S = (1 - \beta)m$ is the number of source examples, $m_T = \beta m$ is the number of target examples, $D = \sqrt{d}/A$, and

$$\nu = \frac{m_T}{m_T + m_S}\left(1 + \frac{m_S}{\sqrt{D^2(m_S + m_T) - m_S m_T}}\right) \tag{8}$$

$$A = \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{4}{\delta})}{m'}} + \lambda \tag{9}$$

$$B = 4\sqrt{\frac{2d\log(2(m+1)) + 2\log(\frac{8}{\delta})}{m}} \tag{10}$$

6.3 Discussion

6.3.1 Unsupervised. Equation 2 indicates that if the optimal predictor error λ^* on both source and target data is large, then there is no good hypothesis from training on the source domain that will work well on the target domain [11, 241]. However, as is more common in the application of domain adaptation, if λ^* is small, then the bound depends on the source error and the $\mathcal{H}\Delta\mathcal{H}$-divergence [11]. The domain-invariant feature learning methods discussed in Section 3.1 try minimizing these two terms [241]: the source error via a task loss on labeled source data and divergence via a divergence measure such as MMD, with reconstruction, or adversarially. While Section 5 shows that on many datasets these methods work, there is no guarantee that such adaptation will increase performance (these are upper bounds), as shown by simple counterexamples [241]. It may actually decrease performance if the marginal label distributions differ significantly between source and target [241].

Equation 3 shows that the target error upper bound alternatively involves the marginal distributions and Equation 4 shows that the lower bound does too. These indicate the importance of aligning the label distributions. If the marginal label distributions are significantly different, then minimizing the source error and divergence between feature representations will actually increase the error [241]. Thus over-training domain-invariant feature learning methods can increase target error, and Zhao et al. [241] experimentally verified this. They found on MNIST, USPS, and SVHN adaptation that during training the target accuracy would initially rise rapidly but would eventually decrease again despite increasing source accuracy, an effect even more apparent with larger differences in the marginal label distributions. It is an open problem as to when the label distributions can be aligned without target labels [241].

6.3.2 Semi-Supervised. Equation 6 indicates that when only source or target data is available, that data should be used (as we might expect). If the source and target are the same, then $\alpha^* = \beta$, which implies a uniform weighting of examples. Given enough target data, source data should not be used at all because it might increase the test-time error. Furthermore, without enough source data using it may also not be worthwhile, i.e., $\alpha^* \approx 0$ [11]. In this paper we focus on unsupervised domain adaptation, but these are important considerations if target labels can be obtained. For example, this shows that it may be better to perform semi-supervised adaptation if some labeled target examples are available rather than using the labeled target examples to hyperparameter tune an unsupervised adaptation method.

7 APPLICATIONS

Domain adaptation has been applied in a variety of areas including computer vision, natural language processing, and for time-series data. Using domain adaptation in these various problems can save the human time that would be spent labeling the target data. In some cases such as image semantic segmentation, providing ground truth is very labor intensive. Each pixel-level annotated image in the Cityscapes dataset took on average 1.5 hours to complete [39]. In addition, similar methods as described in this paper have been applied to the related problem of domain generalization and some other problems as well.

7.1 Computer Vision

Most of the methods surveyed in this paper are for computer vision tasks such as adapting a model trained on synthetic images to real photos (e.g., from synthetic numbers or signs, Table 2), stock photos to real photos (e.g., Amazon to DSLR on the Office dataset, Table 3), or simple to complex images (e.g., MNIST to SVHN, Table 2). Others have been used in robotics for robot grasping [17], autonomous navigation [229], and lifelong learning [222], for semantic segmentation [34, 85, 89, 116, 132, 183, 208, 211, 249] including when additional information is available from a simulator [116], in a medical context for chest X-ray segmentation [28], 3D CT scans to X-ray segmentation [238], MRI to CT scan segmentation [29], and MRI segmentation [161], in low resource situations (where there are very few target data points) [87], in situations with different label sets for each domain [195], for object detection [33, 84, 92], and for person re-identification [9, 46, 61, 218].

7.2 Natural Language Processing

Domain adaptation has been used in natural language processing such as for sentiment analysis (Table 4, [236, 243]), other text classification [123, 236] including weakly-supervised aspect-transfer from one aspect of a dataset to another [236], relation extraction [58], semi-supervised sequence labeling [44], semi-supervised question answering [226], sentence specificity [104], and neural machine translation [20, 27, 36].

7.3 Time Series

For time-series data, domain adaptation has been used for learning temporal latent relationships in health data across different population age groups [165] and to perform speech recognition [87, 191, 243]. In a method addressing the related problem of domain generalization, time-series radio data was used for sleep-stage classification [245].

7.4 Domain Generalization

Domain-invariant feature learning approaches similar to those discussed in Section 3.1 have been used for the related problem of domain generalization, where there are multiple source domains and an unseen target domain [150]. Zhao et al. [245] use an adversarial approach with a domain classifier to learn a model on a dataset collected from a number of people sleeping in various environments that will generalize well to new people and/or new environments (e.g., sleeping in a different room). Ghifary et al. [63] use a reconstruction approach with a denoising autoencoder to improve object recognition generalizability, where the “noise” is different views (domains) of the data (e.g., rotation, change in size, or variation in lighting) and the autoencoder tries to reconstruct corresponding views of the object in other domains. Carlucci et al. [24] propose an adversarial approach combining domain adaptation and generalization while also doing domain mapping.

7.5 Other Problems

Adversarial losses like those used in adversarial domain adaptation methods have also been applied in multiple other settings. Wang et al. [215] created an adversarial spacial dropout network to add occlusions to images to improve the accuracy of object detection algorithms. They also created an adversarial spatial transformer network to add deformations such as rotations to objects to again increase object detection accuracy. Pinto et al. [164] used adversarial agents to improve a robot’s ability to grasp an object via self-supervised learning by employing both shaking and snatching adversaries. Giu et al. [73] used an adversarial loss to predict and demonstrate (i.e., robot will copy) human motion. Rippel et al. [170, 171] used a reconstruction and adversarial loss with an autoencoder for learning higher quality image compression at low bit rates. Sinclair [194] applied adversarial loss to clone a physical model for real-time sound synthesis. Adversarial techniques may also be applied to machine learning security, where the goal is to train a classifier robust to adversarial examples [90, 144].

8 RESEARCH DIRECTIONS

As we have seen, the rapidly-growing body of research focused on unsupervised deep domain adaptation now encompasses many novel methods and components. Here we look at what could be explored in future research to further enhance this existing work.

8.1 Bi-Directional Adaptation

The more difficult domain adaptation problems are far from being solved. Tables 2 through 5 indicate that some domain adaptation problems are harder than others and point to the challenge that more work needs to be focused on these harder problems. While accuracy for SVHN→MNIST ranges from 70.7% to 99.3%, for the reverse case of MNIST→SVHN, the highest without highly-problem-specific hyperparameter tuning is 81.7% by Kumar et al. [108] (though tuned on a small amount of labeled target data). This indicates how this reverse problem is much harder [56, 60]. As a result, few papers offer results for this direction. French et al. [56] were able to vastly improve performance up to 97.0%; however, this required developing a problem-specific unsupervised hyperparameter tuning method. Other methods may similarly benefit from such tuning. Continued work is needed to strengthen general-purpose bi-directional adaptation.

8.2 Hyperparameter Tuning

Some methods such as reverse validation and a problem-specific pixel intensity matching have been applied to hyperparameter tuning without requiring target labels (Section 4.7). While the reverse validation method appears promising, it was not used in most of the methods surveyed (only [61, 159, 163]). This may be because of the increase in computation cost [161] or problems with the reverse validation accuracy not aligning with test accuracy [19]. It is also possible researchers may just be unaware of the method since in the surveyed papers few mention the idea (only [19, 61, 159, 161, 163]). Problem-specific methods such as matching pixel intensity between domains as done by French et al. [56] are possible given some domain knowledge, but hyperparameter tuning methodologies should be developed that will work across a wider range of problems. This remains an open area of research.

8.3 Combining Promising Methods

French et al. [56], Co-DA [108], CAN [99], AutoDIAL [23], Generate to Adapt [182], and WDGRL [190] are promising approaches based on Tables 2 through 4. French et al. uses a student and teacher network for self-ensembling, Co-DA trains multiple (e.g., two) adaptation networks while requiring diversity and agreement in addition to incorporating virtual adversarial training, CAN alternates between clustering and adaptation through minimizing intra-class discrepancy and maximizing inter-class margin, AutoDIAL adjusts batch normalization layer weights, Generate to Adapt uses an embedding-conditional GAN for adversarial domain adaptation, and WDGRL performs adversarial domain adaptation similar to DANN by using a domain classifier. These are largely independent ideas that if combined may result in additional performance gains.

For instance, the student network in French et al. that accepts either a source or target augmented image could be replaced by the AutoDIAL network to learn how much adaptation to perform at each level of the network. Or to combine with adversarial methods, the student and teacher networks' outputs (or an intermediate layer's outputs, as is being explored by Wang et al. [212]) could be fed to a gradient reversal layer followed by a domain classifier, in effect adding an adversarial loss term to the existing two terms used by French et al. Or since French et al. is based upon data augmentation, one might try replacing the existing stochastic data augmentation with a GAN since a GAN can be used for data augmentation (given enough unlabeled training data).

8.4 Balancing Classes

In order to obtain high accuracy on the challenging problem of MNIST→SVHN, French et al. [56] include an additional class-balance term in their loss function, which both improved training stability and helped the network avoid a degenerate local minimum. Though, this term was not required in their other experiments. Clearly, class balancing is an important concern; although, this depends on the dataset being used. Other methods may similarly benefit from balancing classes.

For instance, Hoffman et al. [83] note that the frequency-weighted intersection over union results in their paper were very close to the target-only model accuracy (an approximate upper bound). Thus, they conclude that domain mapping followed by domain-invariant feature learning is very effective for the common classes in the SYNTHIA dataset (season adaptation on a synthetic driving dataset). It is possible then that additional balancing of classes could help the not-as-common classes to perform better. In addition, data augmentation through occluding parts of the images may improve class balancing as would the adversarial spatial dropout network by Wang et al. [215] since the two best classes (road and sky) were likely in almost every image.

8.5 Incorporating Improved Image-to-Image Translation Methods

Bousmalis et al. [18] with PixelDA had difficulty applying their method with large domain differences. However, other image-to-image translation methods like XGAN [172] have been developed that may support larger domain shifts. These methods could be extended to domain adaptation directly or also incorporating a semantic consistency loss (as explained in Section 4.1). This may allow for more substantial differences between domains. Similarly, image-to-image translation methods like StarGAN [35] have been developed for multiple domains, which could be extended for multi-domain adaptation.

8.6 Futher Experimental Comparison Between Methods

As shown in Table 2, French et al. [56] outperforms all the other methods and Co-DA [108] is quite close behind (with the advantage that it does not require highly-problem-specific tuning on MNIST→SVHN). In Table 3, CAN [99] outperforms the others followed by Generate to Adapt [182]. Finally, in Table 4, WDGRL [190] generally performs the best. However, these methods are not all compared on the same dataset, making a direct comparison difficult. Additional experiments must be performed to see how these methods compare. Similarly, other promising approaches may outperform other methods on some datasets, which could be determined through additional experiments.

These comparisons can be made easier through developing a unified implementation of these various methods. Schneider et al. [186] are developing such an open-source set of implementations of state-of-the-art domain adaptation (and domain generalization) methods. The results provided in individual papers have different hyperparameters, data augmentation, network architectures, etc. that can make direct comparisons challenging. Using a unified implementation of these methods can facilitate more clearly understanding what aspects of a method are responsible for performance gains and also support combining the novel elements from multiple methods.

8.7 Limitations of Datasets

Varying amounts of source and target data are available in different situations. The datasets used for comparisons (the image datasets listed in Table 5 and the Amazon review dataset) are relatively small when compared with the sizes of datasets commonly in use in deep learning, e.g., ImageNet [45, 175] (though ImageNet is often used to pretrain adaptation networks). For example, Sankaranarayanan et al. [182] note how GANs require a lot of training data. This may limit GAN-based methods from being used on too small of source or target datasets. Modifications may need to be developed for such low resource situations, an area explored by Hosseini-Asl et al. [87]. Additionally, most domain adaptation datasets are for computer vision. To spur research in other application areas, other datasets could be created.

8.8 Other Applications

Other application areas may benefit from performing domain adaptation as have those discussed in Section 7. In particular, only a few methods were applied to time-series data. One time-series application that may benefit from adaptation is activity prediction, e.g., adapting from one type of sensor to another or from one person's data to another's. Some added challenges in this context may be the large differences in feature spaces due to the wide variety of sensors used (e.g., an event stream of fixed motion sensors turning on and off in a smart home vs. sampled motion and location data collected from smart phones or watches) or the difference in labels (e.g., one model may learn a "walk" activity while another learns "exercise" or may learn "read" while another model learns "school"). Applying domain adaptation in new areas may yield novel methods or components applicable in other areas as well.

9 CONCLUSIONS

For supervised learning, deep neural networks are in prevalent use, but these networks require large labeled datasets for training. Unsupervised domain adaptation can be used to adapt deep networks to possibly-smaller datasets that may not even have target labels. Several categories of methods have been developed for this goal: domain-invariant feature learning, domain mapping, normalization statistics-based, and ensemble-based methods. These various methods have some unique and common elements as we have discussed. Additionally, theoretical results provide some insight into empirical observations. Some methods appear very promising, but further research is required for direct comparisons, novel method combinations, improved bi-directional adaptation, and use for novel datasets and applications.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 1543656 and 1734558.

REFERENCES

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. 2014. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446* (2014).

[2] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. 2018. Unsupervised Attention-guided Image-to-Image Translation. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 3693–3703. http://papers.nips.cc/paper/7627-unsupervised-attention-guided-image-to-image-translation.pdf

[3] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. 2018. ComboGAN: Unrestrained Scalability for Image Domain Translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 214–223. http://proceedings.mlr.press/v70/arjovsky17a.html

[5] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and Equilibrium in Generative Adversarial Nets (GANs). In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 224–232. http://proceedings.mlr.press/v70/arora17a.html

[6] Sanjeev Arora, Andrej Risteski, and Yi Zhang. 2018. Do GANs learn the distribution? Some Theory and Empirics. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJehNfW0-

[7] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. 2019. There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkgKBhA5Y7

[8] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[9] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. 2018. Domain Adaptation through Synthesis for Unsupervised Person Re-identification. In *The European Conference on Computer Vision (ECCV)*.

[10] Oscar Beijbom. 2012. Domain adaptations for computer vision applications. *arXiv preprint arXiv:1211.4860* (2012).

[11] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning* 79, 1 (01 May 2010), 151–175. https://doi.org/10.1007/s10994-009-5152-4

[12] Sagie Benaim and Lior Wolf. 2017. One-Sided Unsupervised Domain Mapping. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 752–762. http://papers.nips.cc/paper/6677-one-sided-unsupervised-domain-mapping.pdf

[13] David Berthelot, Tom Schumm, and Luke Metz. 2017. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717* (2017).

[14] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. 2017. Demystifying MMD GANs. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lUOzWCW

[15] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 440–447.

[16] Ali Borji. 2018. Pros and Cons of GAN Evaluation Measures. *arXiv preprint arXiv:1802.03446* (2018).

[17] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke. 2018. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 4243–4250. https://doi.org/10.1109/ICRA.2018.8460875

[18] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. 2017. Unsupervised Pixel-Level Domain Adaptation With Generative Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain Separation Networks. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 343–351. http://papers.nips.cc/paper/6254-domain-separation-networks.pdf

[20] Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*. 118–126.

[21] Lars Bungum and Björn Gambäck. 2011. A survey of domain adaptation in machine translation: Towards a refinement of domain space. In *Proceedings of the India-Norway Workshop on Web Concepts and Technologies*, Vol. 112.

[22] Jinming Cao, Oren Katzir, Peng Jiang, Dani Lischinski, Danny Cohen-Or, Changhe Tu, and Yangyan Li. 2018. Dida: Disentangled synthesis for domain adaptation. *arXiv preprint arXiv:1805.08019* (2018).

[23] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. 2017. Autodial: Automatic domain alignment layers. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 5077–5085.

[24] Fabio M Carlucci, Paolo Russo, Tatiana Tommasi, and Barbara Caputo. 2018. Agnostic Domain Generalization. *arXiv preprint arXiv:1808.01102* (2018).

[25] Rich Caruana. 1997. Multitask Learning. *Machine Learning* 28, 1 (01 Jul 1997), 41–75. https://doi.org/10.1023/A:1007379606734

[26] Olivier Chapelle and Alexander Zien. 2005. Semi-supervised classification by low density separation.. In *AISTATS*, Vol. 2005. Citeseer, 57–64.

[27] Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*. 40–46.

[28] Cheng Chen, Qi Dou, Hao Chen, and Pheng-Ann Heng. 2018. Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation in Chest X-Ray Segmentation. In *Machine Learning in Medical Imaging*, Yinghuan Shi, Heung-Il Suk, and Mingxia Liu (Eds.). Springer International Publishing, Cham, 143–151.

[29] Cheng Chen, Qi Dou, Hao Chen, Jing Qin, and Pheng-Ann Heng. 2019. Synergistic Image and Feature Adaptation: Towards Cross-Modality Domain Adaptation for Medical Image Segmentation. *arXiv preprint arXiv:1901.08211* (2019).

[30] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized Denoising Autoencoders for Domain Adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12) (ICML '12)*, John Langford and Joelle Pineau (Eds.). Omnipress, New York, NY, USA, 767–774.

[31] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2172–2180. http://papers.nips.cc/paper/6399-infogan-interpretable-representation-learning-by-information-maximizing-generative-adversarial-nets.pdf

[32] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. 2018. Attention-GAN for Object Transfiguration in Wild Images. In *The European Conference on Computer Vision (ECCV)*.

[33] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. 2018. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[34] Yuhua Chen, Wen Li, and Luc Van Gool. 2018. ROAD: Reality Oriented Adaptation for Semantic Segmentation of Urban Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[35] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[36] Chenhui Chu and Rui Wang. 2018. A Survey of Domain Adaptation for Neural Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1304–1319.

[37] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2980–2988. http://papers.nips.cc/paper/5653-a-recurrent-latent-variable-model-for-sequential-data.pdf

[38] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. 2013. Transfer learning for activity recognition: a survey. *Knowledge and Information Systems* 36, 3 (01 Sep 2013), 537–556. https://doi.org/10.1007/s10115-013-0665-3

[39] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[40] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3730–3739. http://papers.nips.cc/paper/6963-joint-distribution-optimal-transportation-for-domain-adaptation.pdf

[41] Gabriela Csurka. 2017. A comprehensive survey on domain adaptation for visual applications. In *Domain Adaptation in Computer Vision Applications*. Springer, 1–35.

[42] Gabriela Csurka. 2017. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374* (2017).

[43] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. 2018. DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 467–483.

[44] Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 256–263.

[45] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[46] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. 2018. Image-Image Domain Adaptation With Preserved Self-Similarity and Domain-Dissimilarity for Person Re-Identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[47] Zhijie Deng, Yucen Luo, and Jun Zhu. 2019. Cluster Alignment with a Teacher for Unsupervised Domain Adaptation. *arXiv preprint arXiv:1903.09980* (2019).

[48] Emily L Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 1486–1494. http://papers.nips.cc/paper/5773-deep-generative-image-models-using-a-laplacian-pyramid-of-adversarial-networks.pdf

[49] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2017. Adversarial feature learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJtNZAFgg

[50] Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning* 79, 1 (01 May 2010), 123–149. https://doi.org/10.1007/s10994-009-5148-0

[51] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. 2017. Generative Multi-Adversarial Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Byk-VI9eg

[52] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. 2015. Training Generative Neural Networks via Maximum Mean Discrepancy Optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI'15)*. AUAI Press, Arlington, Virginia, United States, 258–267. http://dl.acm.org/citation.cfm?id=3020847.3020875

[53] M. El Habib Daho, N. Settouti, M. E. A. Lazouni, and M. E. A. Chikh. 2014. Weighted vote for trees aggregation in Random Forest. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*. 438–443. https:

//doi.org/10.1109/ICMCS.2014.6911187

[54] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian Goodfellow. 2018. Many Paths to Equilibrium: GANs Do Not Need to Decrease a Divergence At Every Step. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByQpn1ZA-

[55] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852* (2016).

[56] Geoff French, Michal Mackiewicz, and Mark Fisher. 2018. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkpoTaxA-

[57] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. 2018. Geometry-Consistent Adversarial Networks for One-Sided Unsupervised Domain Mapping. *arXiv preprint arXiv:1809.05852* (2018).

[58] Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Vol. 2. 425–429.

[59] Zhe Gan, Liqun Chen, Weiyao Wang, Yuchen Pu, Yizhe Zhang, Hao Liu, Chunyuan Li, and Lawrence Carin. 2017. Triangle Generative Adversarial Networks. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5247–5256. http://papers.nips.cc/paper/7109-triangle-generative-adversarial-networks.pdf

[60] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, 1180–1189. http://proceedings.mlr.press/v37/ganin15.html

[61] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* 17, 59 (2016), 1–35. http://jmlr.org/papers/v17/15-239.html

[62] Jon Gauthier. 2014. Conditional generative adversarial nets for convolutional face generation.

[63] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. 2015. Domain Generalization for Object Recognition With Multi-Task Autoencoders. In *The IEEE International Conference on Computer Vision (ICCV)*.

[64] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 597–613.

[65] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. 2018. Unsupervised Multi-Target Domain Adaptation: An Information Theoretic Approach. *arXiv preprint arXiv:1810.11547* (2018).

[66] Arnab Ghosh, Viveka Kulharia, Vinay P Namboodiri, Philip HS Torr, and Puneet K Dokania. 2018. Multi-agent diverse generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8513–8521.

[67] Ian Goodfellow. 2016. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* (2016).

[68] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[69] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[70] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. 2007. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*. 513–520.

[71] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, Mar (2012), 723–773.

[72] Aditya Grover, Manik Dhar, and Stefano Ermon. 2017. Flow-GAN: Combining maximum likelihood and adversarial learning in generative models. *arXiv preprint arXiv:1705.08868* (2017).

[73] Liangyan Gui, Kevin Zhang, Yu-Xiong Wang, Xiaodan Liang, José MF Moura, and Manuela M Veloso. 2018. Teaching Robots to Predict Human Motion. (2018). preprint on webpage at http://www.cs.cmu.edu/~mmv/papers/18iros-GuiEtAl.pdf.

[74] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5767–5777. http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf

[75] Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-Source Domain Adaptation with Mixture of Experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4694–4703.

[76] Eman T Hassan, Xin Chen, and David Crandall. 2018. Unsupervised Domain Adaptation using Generative Models and Self-ensembling. *arXiv preprint arXiv:1812.00479* (2018).

[77] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6626–6637. http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.pdf

[78] Avinash Hindupur. 2018. The GAN Zoo. Retrieved February 25, 2019 from https://github.com/hindupuravinash/the-gan-zoo

[79] Saifuddin Hitawala. 2018. Comparative Study on Generative Adversarial Networks. *arXiv preprint arXiv:1801.04271* (2018).

[80] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. 2018. MGAN: Training Generative Adversarial Nets with Multiple Generators. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkmu5b0a-

[81] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://doi.org/10.1162/neco.1997.9.8.1735

[82] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. 2018. Algorithms and Theory for Multiple-Source Adaptation. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 8246–8256. http://papers.nips.cc/paper/8046-algorithms-and-theory-for-multiple-source-adaptation.pdf

[83] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 1994–2003. http://proceedings.mlr.press/v80/hoffman18a.html

[84] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649* (2016).

[85] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. 2018. Conditional Generative Adversarial Network for Structured Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[86] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. 2019. How Generative Adversarial Networks and Their Variants Work: An Overview. *ACM Comput. Surv.* 52, 1, Article 10 (Feb. 2019), 43 pages. https://doi.org/10.1145/3301282

[87] Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. 2019. Augmented Cyclic Adversarial Learning for Low Resource Domain Adaptation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1G9doA9F7

[88] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. 2018. Learning to cluster in order to transfer across domains and tasks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByRWCqvT-

[89] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. 2018. Domain transfer through deep activation matching. In *The European Conference on Computer Vision (ECCV)*.

[90] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. 2011. Adversarial Machine Learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (AISec '11)*. ACM, New York, NY, USA, 43–58. https://doi.org/10.1145/2046684.2046692

[91] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-Image Translation. *arXiv preprint arXiv:1804.04732* (2018).

[92] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Cross-Domain Weakly-Supervised Object Detection Through Progressive Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[93] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 448–456. http://proceedings.mlr.press/v37/ioffe15.html

[94] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[95] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407* (2018).

[96] Jing Jiang. 2008. *Domain adaptation in natural language processing*. Technical Report. University of Illinois at Urbana-Champaign.

[97] Alexia Jolicoeur-Martineau. 2018. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734* (2018).

[98] Mahesh Joshi, William W. Cohen, Mark Dredze, and Carolyn P. Rosé. 2012. Multi-domain Learning: When Do Domains Matter?. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1302–1312. http://dl.acm.org/citation.cfm?id=2390948.2391096

[99] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. 2019. Contrastive Adaptation Network for Unsupervised Domain Adaptation. *arXiv preprint arXiv:1901.00976* (2019).

[100] Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. 2018. Deep Adversarial Attention Alignment for Unsupervised Domain Adaptation: the Benefit of Target Expectation Maximization. In *The European Conference on Computer Vision (ECCV)*.

[101] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Hk99zCeAb

[102] Mahyar Khayatkhoei, Maneesh K. Singh, and Ahmed Elgammal. 2018. Disconnected Manifold Learning for Generative Adversarial Networks. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7343–7353. http://papers.nips.cc/paper/7964-disconnected-manifold-learning-for-generative-adversarial-networks.pdf

[103] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to Discover Cross-domain Relations with Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, 1857–1865. http://dl.acm.org/citation.cfm?id=3305381.3305573

[104] Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. 2018. Domain Agnostic Real-Valued Specificity Prediction. *arXiv preprint arXiv:1811.05085* (2018).

[105] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. 2017. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215* (2017).

[106] Wouter M Kouw. 2018. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806* (2018).

[107] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images.* Technical Report.

[108] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. 2018. Co-regularized Alignment for Unsupervised Domain Adaptation. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 9345–9356. http://papers.nips.cc/paper/8146-co-regularized-alignment-for-unsupervised-domain-adaptation.pdf

[109] Samuli Laine and Timo Aila. 2017. Temporal Ensembling for Semi-Supervised Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJ6oOfqge

[110] Issam Laradji and Reza Babanezhad. 2018. M-ADDA: Unsupervised Domain Adaptation with Deep Metric Learning. *arXiv preprint arXiv:1807.02552* (2018).

[111] Alessandro Lazaric. 2012. *Transfer in Reinforcement Learning: A Framework and a Survey.* Springer Berlin Heidelberg, Berlin, Heidelberg, 143–173. https://doi.org/10.1007/978-3-642-27645-3_5

[112] Trung Le, Khanh Nguyen, and Dinh Phung. 2018. Theoretical Perspective of Deep Domain Adaptation. *arXiv preprint arXiv:1811.06199* (2018).

[113] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. 1998. The MNIST database of handwritten digits. Retrieved August 16, 2018 from http://yann.lecun.com/exdb/mnist/

[114] Y. LeCun, O. Matan, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jacket, and H. S. Baird. 1990. Handwritten zip code recognition with multilayer networks. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, Vol. ii. 35–40 vol.2. https://doi.org/10.1109/ICPR.1990.119325

[115] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. 2019. Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation. *arXiv preprint arXiv:1903.04064* (2019).

[116] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. 2019. SPIGAN: Privileged Adversarial Learning from Simulation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkxoNnC5FQ

[117] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. 2017. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*. 2203–2213.

[118] Jerry Li. 2018. Twin-GAN–Unpaired Cross-Domain Image Translation with Weight-Sharing GANs. *arXiv preprint arXiv:1809.00946* (2018).

[119] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P Xing. 2018. Semantic-aware grad-gan for virtual-to-real urban scene adaption. *arXiv preprint arXiv:1801.01726* (2018).

[120] Yujia Li, Kevin Swersky, and Rich Zemel. 2015. Generative Moment Matching Networks. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 1718–1727. http://proceedings.mlr.press/v37/li15.html

[121] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. 2018. Adaptive Batch Normalization for practical domain adaptation. *Pattern Recognition* 80 (2018), 109 – 117. https://doi.org/10.1016/j.patcog.2018.03.005

[122] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled Generative Adversarial Networks. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 469–477. http://papers.nips.cc/paper/6544-coupled-generative-adversarial-networks.pdf

[123] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1–10.

[124] Shaohui Liu, Yi Wei, Jiwen Lu, and Jie Zhou. 2018. An Improved Evaluation Framework for Generative Adversarial Networks. *arXiv preprint arXiv:1803.07474* (2018).

[125] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *The IEEE International Conference on Computer Vision (ICCV)*.

[126] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, 97–105. http://proceedings.mlr.press/v37/long15.html

[127] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional Adversarial Domain Adaptation. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 1640–1650. http://papers.nips.cc/paper/7436-conditional-adversarial-domain-adaptation.pdf

[128] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. 2013. Transfer Feature Learning with Joint Distribution Adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*.

[129] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised Domain Adaptation with Residual Transfer Networks. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 136–144. http://papers.nips.cc/paper/6110-unsupervised-domain-adaptation-with-residual-transfer-networks.pdf

[130] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 2208–2217. http://proceedings.mlr.press/v70/long17a.html

[131] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. 2015. Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems* 80 (2015), 14 – 23. https://doi.org/10.1016/j.knosys.2015.01.010 25th anniversary of Knowledge-Based Systems.

[132] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. 2018. Taking A Closer Look at Domain Shift: Category-level Adversaries for Semantics Consistent Domain Adaptation. *arXiv preprint arXiv:1809.09478* (2018).

[133] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).

[134] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota BulÃš, Barbara Caputo, and Elisa Ricci. 2018. Boosting Domain Adaptation by Discovering Latent Domains. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[135] P Manisha and Sujit Gujar. 2018. Generative Adversarial Networks (GANs): What it can generate and What it cannot? *arXiv preprint arXiv:1804.00140* (2018).

[136] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain Adaptation with Multiple Sources. In *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.). Curran Associates, Inc., 1041–1048. http://papers.nips.cc/paper/3550-domain-adaptation-with-multiple-sources.pdf

[137] Xudong Mao and Qing Li. 2018. Unpaired Multi-domain Image Generation via Regularized Conditional GANs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. AAAI Press, 2553–2559. http://dl.acm.org/citation.cfm?id=3304889.3305015

[138] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2794–2802.

[139] Anna Margolis. 2011. A Literature Review of Domain Adaptation with Unlabeled Data.

[140] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. 2017. Unrolled generative adversarial networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BydrOIcle

[141] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[142] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=

B1QRgziT-

[143] Takeru Miyato and Masanori Koyama. 2018. cGANs with Projection Discriminator. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByS1VpgRZ

[144] T. Miyato, S. Maeda, S. Ishii, and M. Koyama. 2018. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), 1–1. https://doi.org/10.1109/TPAMI.2018.2858821

[145] Boris Moiseev, Artem Konev, Alexander Chigorin, and Anton Konushin. 2013. Evaluation of Traffic Sign Recognition Methods Trained on Synthetically Generated Data. In *Advanced Concepts for Intelligent Vision Systems*, Jacques Blanc-Talon, Andrzej Kasinski, Wilfried Philips, Dan Popescu, and Paul Scheunders (Eds.). Springer International Publishing, Cham, 576–583.

[146] Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris* (1781).

[147] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. 2018. Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators. *arXiv preprint arXiv:1807.11346* (2018).

[148] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. 2018. Minimal-Entropy Correlation Alignment for Unsupervised Deep Domain Adaptation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJWechg0Z

[149] Pietro Morerio and Vittorio Murino. 2017. Correlation Alignment by Riemannian Metric for Domain Adaptation. *arXiv preprint arXiv:1705.08180* (2017).

[150] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain Generalization via Invariant Feature Representation. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. PMLR, 10–18. http://proceedings.mlr.press/v28/muandet13.html

[151] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, Vol. 2011. 5.

[152] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[153] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 271–279. http://papers.nips.cc/paper/6066-f-gan-training-generative-neural-samplers-using-variational-divergence-minimization.pdf

[154] Augustus Odena, Jacob Buckman, Catherine Olsson, Tom Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. 2018. Is Generator Conditioning Causally Related to GAN Performance?. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 3846–3855. http://proceedings.mlr.press/v80/odena18a.html

[155] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional Image Synthesis with Auxiliary Classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 2642–2651. http://proceedings.mlr.press/v70/odena17a.html

[156] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (Oct 2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[157] David Keetae Park, Seungjoo Yoo, Hyojin Bahng, Jaegul Choo, and Noseong Park. 2018. MEGAN: Mixture of Experts of Generative Adversarial Networks for Multimodal Image Generation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. AAAI Press, 878–884. http://dl.acm.org/citation.cfm?id=3304415.3304540

[158] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. 2015. Visual Domain Adaptation: A survey of recent advances. *IEEE Signal Processing Magazine* 32, 3 (May 2015), 53–69. https://doi.org/10.1109/MSP.2014.2347059

[159] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2018. Multi-adversarial domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[160] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2018. Moment Matching for Multi-Source Domain Adaptation. *arXiv preprint arXiv:1812.01754* (2018).

[161] Christian S Perone, Pedro Ballester, Rodrigo C Barros, and Julien Cohen-Adad. 2018. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *arXiv preprint arXiv:1811.06042* (2018).

[162] David Pfau and Oriol Vinyals. 2016. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945* (2016).

[163] Pedro O. Pinheiro. 2018. Unsupervised Domain Adaptation With Similarity Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[164] L. Pinto, J. Davidson, and A. Gupta. 2017. Supervision via competition: Robot adversaries for learning tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA).* 1601–1608. https://doi.org/10.1109/ICRA.2017.7989190

[165] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. 2017. Variational adversarial deep domain adaptation for health care time series analysis. In *International Conference on Learning Representations.* https://openreview.net/forum?id=rk9eAFcxg

[166] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[167] J. Rapin and O. Teytaud. 2018. Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad.

[168] Ievgen Redko, Amaury Habrard, and Marc Sebban. 2017. Theoretical Analysis of Domain Adaptation withÂăOptimal Transport. In *Machine Learning and Knowledge Discovery in Databases*, Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski (Eds.). Springer International Publishing, Cham, 737–753.

[169] Jian Ren, Jianchao Yang, Ning Xu, and David J Foran. 2018. Factorized Adversarial Networks for Unsupervised Domain Adaptation. *arXiv preprint arXiv:1806.01376* (2018).

[170] Oren Rippel and Lubomir Bourdev. 2017. Real-Time Adaptive Image Compression. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 2922–2930. http://proceedings.mlr.press/v70/rippel17a.html

[171] Oren Rippel, Lubomir Bourdev, Carissa Lew, and Sanjay Nair. 2018. Using generative adversarial networks in compression. US Patent App. 15/844,449.

[172] Amélie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy. 2017. XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings. *arXiv preprint arXiv:1711.05139* (2017).

[173] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. 2018. Residual Parameter Transfer for Deep Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[174] A. Rozantsev, M. Salzmann, and P. Fua. 2019. Beyond Sharing Weights for Deep Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 4 (April 2019), 801–814. https://doi.org/10.1109/TPAMI.2018.2814042

[175] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[176] Paolo Russo, Fabio M. Carlucci, Tatiana Tommasi, and Barbara Caputo. 2018. From Source to Target and Back: Symmetric Bi-Directional Adaptive GAN. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[177] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting Visual Category Models to New Domains. In *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 213–226.

[178] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Asymmetric Tri-training for Unsupervised Domain Adaptation. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 2988–2997. http://proceedings.mlr.press/v70/saito17a.html

[179] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. 2018. Adversarial Dropout Regularization. In *International Conference on Learning Representations.* https://openreview.net/forum?id=HJIoJWZCZ

[180] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[181] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2234–2242. http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[182] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo, and Rama Chellappa. 2018. Generate to Adapt: Aligning Domains Using Generative Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[183] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. 2018. Learning From Synthetic Data: Addressing Domain Shift for Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[184] Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. 2018. A Classification-Based Study of Covariate Shift in GAN Distributions. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine*

Learning Research), Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 4487–4496. http://proceedings.mlr.press/v80/santurkar18a.html

[185] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. 2018. How Does Batch Normalization Help Optimization? In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 2483–2493. http://papers.nips.cc/paper/7515-how-does-batch-normalization-help-optimization.pdf

[186] Steffen Schneider, Alexander S. Ecker, Jakob H. Macke, and Matthias Bethge. 2018. Salad: A Toolbox for Semi-supervised Adaptive Learning Across Domains. https://openreview.net/forum?id=S1lTifykqm

[187] Alice Schoenauer Sebag, Louise Heinrich, Marc Schoenauer, Michèle Sebag, Lani Wu, and Steven Altschuler. 2019. Multi-Domain Adversarial Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Sklv5iRqYX

[188] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. 2016. Learning Transferrable Representations for Unsupervised Domain Adaptation. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2110–2118. http://papers.nips.cc/paper/6360-learning-transferrable-representations-for-unsupervised-domain-adaptation.pdf

[189] L. Shao, F. Zhu, and X. Li. 2015. Transfer Learning for Visual Categorization: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* 26, 5 (May 2015), 1019–1034. https://doi.org/10.1109/TNNLS.2014.2330900

[190] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[191] Yusuke Shinohara. 2016. Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition. In *Interspeech 2016*. 2369–2372. https://doi.org/10.21437/Interspeech.2016-879

[192] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. 2017. Learning From Simulated and Unsupervised Images Through Adversarial Training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[193] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. 2018. A DIRT-T Approach to Unsupervised Domain Adaptation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1q-TM-AW

[194] Stephen Sinclair. 2018. Sounderfeit: Cloning a Physical Model using a Conditional Adversarial Autoencoder. *arXiv preprint arXiv:1806.09617* (2018).

[195] Kihyuk Sohn, Wenling Shang, Xiang Yu, and Manmohan Chandraker. 2019. Unsupervised Domain Adaptation for Distance Metric Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BklhAj09K7

[196] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. 2011. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*. 1453–1460. https://doi.org/10.1109/IJCNN.2011.6033395

[197] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of Frustratingly Easy Domain Adaptation. In *AAAI Conference on Artificial Intelligence*. https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12443

[198] Baochen Sun and Kate Saenko. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *Computer Vision – ECCV 2016 Workshops*, Gang Hua and Hervé Jégou (Eds.). Springer International Publishing, Cham, 443–450.

[199] Shiliang Sun, Honglei Shi, and Yuanbin Wu. 2015. A survey of multi-source domain adaptation. *Information Fusion* 24 (2015), 84 – 92. https://doi.org/10.1016/j.inffus.2014.12.003

[200] Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. 2016. Generative Models and Model Criticism via Optimized Maximum Mean Discrepancy. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJWHIKqgl

[201] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[202] Yaniv Taigman, Adam Polyak, and Lior Wolf. 2016. Unsupervised Cross-Domain Image Generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Sk2Im59ex

[203] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A Survey on Deep Transfer Learning. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis (Eds.). Springer International Publishing, Cham, 270–279.

[204] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 1195–1204. http://papers.nips.cc/paper/6719-mean-teachers-are-better-role-models-weight-averaged-consistency-targets-improve-semi-supervised-deep-learning-results.pdf

[205] Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, Jul (2009), 1633–1685.

[206] Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2016. A note on the evaluation of generative models. In *International Conference on Learning Representations*. https://arxiv.org/abs/1511.01844

[207] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann SIMON-GABRIEL, and Bernhard Schölkopf. 2017. AdaGAN: Boosting Generative Models. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5424–5433. http://papers.nips.cc/paper/7126-adagan-boosting-generative-models.pdf

[208] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. 2018. Learning to Adapt Structured Output Space for Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[209] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. 2015. Simultaneous Deep Transfer Across Domains and Tasks. In *The IEEE International Conference on Computer Vision (ICCV)*.

[210] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[211] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. 2018. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. *arXiv preprint arXiv:1811.12833* (2018).

[212] Jiawei Wang, Zhaoshui He, Chengjian Feng, Zhouping Zhu, Qinzhuang Lin, Jun Lv, and Shengli Xie. 2018. Domain Confusion with Self Ensembling for Unsupervised Adaptation. *arXiv preprint arXiv:1810.04472* (2018).

[213] Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312 (2018), 135 – 153. https://doi.org/10.1016/j.neucom.2018.05.083

[214] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. 2019. Transferable Attention for Domain Adaptation. (2019).

[215] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. 2017. A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[216] Yifei Wang, Wen Li, Dengxin Dai, and Luc Van Gool. 2017. Deep Domain Adaptation by Geodesic Distance Minimization. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

[217] Kai-Ya Wei and Chiou-Ting Hsu. 2018. Generative Adversarial Guided Learning for Domain Adaptation. *British Machine Vision Conference* (2018).

[218] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. 2018. Person Transfer GAN to Bridge Domain Gap for Person Re-Identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[219] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big Data* 3, 1 (28 May 2016), 9. https://doi.org/10.1186/s40537-016-0043-6

[220] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. 2017. On the quantitative analysis of decoder-based generative models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1M8JF9xx

[221] Yuxin Wu and Kaiming He. 2018. Group Normalization. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 3–19.

[222] M. Wulfmeier, A. Bewley, and I. Posner. 2018. Incremental Adversarial Domain Adaptation for Continually Changing Environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 1–9. https://doi.org/10.1109/ICRA.2018.8460982

[223] Zhao Xin. 2019. A collection of AWESOME things about domian adaptation. Retrieved March 20, 2019 from https://github.com/zhaoxin94/awsome-domain-adaptation

[224] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. 2018. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755* (2018).

[225] Yongxin Yang and Timothy M Hospedales. 2015. A unified perspective on multi-domain and multi-task learning. In *International Conference on Learning Representations*. https://arxiv.org/abs/1412.7489

[226] Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1040–1050.

[227] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. DualGAN: Unsupervised Dual Learning for Image-To-Image Translation. In *The IEEE International Conference on Computer Vision (ICCV)*.

[228] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. 2016. Pixel-level domain transfer. In *European Conference on Computer Vision*. Springer, 517–532.

[229] Jaeyoon Yoo, Yongjun Hong, YungKyun Noh, and Sungroh Yoon. 2017. Domain Adaptation Using Adversarial Learning for Autonomous Navigation. *arXiv preprint arXiv:1712.03742* (2017).

[230] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365* (2015).

[231] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-Attention Generative Adversarial Networks. *arXiv preprint arXiv:1805.08318* (2018).

[232] Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. Generative Warfare Nets: Ensemble via Adversaries and Collaborators.. In *IJCAI*. 3075–3081.

[233] JiChao Zhang. 2019. Adversarial Nets Papers. Retrieved February 25, 2019 from https://github.com/zhangqianhui/AdversarialNetsPapers

[234] Jing Zhang, Wanqing Li, and Philip Ogunbona. 2017. Transfer learning for cross-dataset recognition: a survey. *arXiv preprint arXiv:1705.04396* (2017).

[235] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. 2018. Collaborative and Adversarial Network for Unsupervised Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[236] Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2017. Aspect-augmented Adversarial Networks for Domain Adaptation. *Transactions of the Association for Computational Linguistics* 5 (2017), 515–528. https://doi.org/10.1162/tacl_a_00077 arXiv:https://doi.org/10.1162/tacl_a_00077

[237] Yang Zhang, Philip David, and Boqing Gong. 2017. Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. In *The IEEE International Conference on Computer Vision (ICCV)*.

[238] Yue Zhang, Shun Miao, Tommaso Mansi, and Rui Liao. 2018. Task Driven Generative Modeling for Unsupervised Domain Adaptation: Application to X-ray Image Segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger (Eds.). Springer International Publishing, Cham, 599–607.

[239] Y. Zhang, N. Wang, S. Cai, and L. Song. 2018. Unsupervised Domain Adaptation by Mapped Correlation Alignment. *IEEE Access* 6 (2018), 44698–44706. https://doi.org/10.1109/ACCESS.2018.2865249

[240] Zhen Zhang, Mianzhi Wang, Yan Huang, and Arye Nehorai. 2018. Aligning Infinite-Dimensional Covariance Matrices in Reproducing Kernel Hilbert Spaces for Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[241] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. 2019. On Learning Invariant Representation for Domain Adaptation. *arXiv preprint arXiv:1901.09453* (2019).

[242] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. 2018. Adversarial Multiple Source Domain Adaptation. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 8559–8570. http://papers.nips.cc/paper/8075-adversarial-multiple-source-domain-adaptation.pdf

[243] Han Zhao, Zhenyao Zhu, Junjie Hu, Adam Coates, and Geoff Gordon. 2017. Principled hybrids of generative and discriminative domain adaptation. *arXiv preprint arXiv:1705.09011* (2017).

[244] Junbo Zhao, Michael Mathieu, and Yann LeCun. 2017. Energy-based generative adversarial network. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ryh9pmcee

[245] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S. Jaakkola, and Matt T. Bianchi. 2017. Learning Sleep Stages from Radio Signals: A Conditional Adversarial Architecture. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 4100–4109. http://proceedings.mlr.press/v70/zhao17d.html

[246] Sicheng Zhao, Bichen Wu, Joseph Gonzalez, Sanjit A Seshia, and Kurt Keutzer. 2018. Unsupervised Domain Adaptation: from Simulation Engine to the RealWorld. *arXiv preprint arXiv:1803.09180* (2018).

[247] Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. 2010. Cross Validation Framework to Choose amongst Models and Datasets for Transfer Learning. In *Machine Learning and Knowledge Discovery in Databases*, José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 547–562.

[248] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *The IEEE International Conference on Computer Vision (ICCV)*.

[249] Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. In *The European Conference on Computer Vision (ECCV)*.