IMPROVING ADVERSARIAL DISCRIMINATIVE DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial discriminative domain adaptation (ADDA) is an efficient framework for unsupervised domain adaptation, where the source and target domains are assumed to have the same classes, but no labels are available for the target domain. While ADDA has already achieved better training efficiency and competitive accuracy in comparison to other adversarial based methods, we investigate whether we can improve performance by incorporating task knowledge into the adversarial loss functions. We achieve this by extending the discriminator output over the source classes and leverage on the distribution over the source encoder posteriors, which is fixed during adversarial training, in order to align a shared encoder distribution to the source domain. The shared encoder can receive a proportion of examples from both the source and target datasets, in order to smooth the learned distribution and improve its convergence properties during adversarial training. We additionally consider how the extended discriminator can be regularized in order to further improve performance, by treating the discriminator as a denoising autoencoder and corrupting its input. Our final design employs maximum mean discrepancy and reconstruction-based loss functions for adversarial training. We validate our framework on standard datasets like MNIST, USPS, SVHN, MNIST-M and Office-31. Our results on all datasets show that our proposal is both simple and efficient, as it competes or outperforms the state-of-the-art in unsupervised domain adaptation, whilst offering lower complexity than other recent adversarial methods such as DIFA and CoGAN.

1 INTRODUCTION

The long-standing goal in visual learning is to generalize the learned knowledge from a source domain to new domains, even without the presence of labels in the target domains. Significant strides have been made towards this goal in the last few years, mainly due to proposals based on multilayered convolutional neural networks that have shown cross-domain generalizations and fast learning of new tasks by fine-tuning on limited subsets of labelled data.

Unsupervised domain adaptation directly aims at improving the generalization capability between a labelled source domain and an unlabelled target domain. Deep domain adaptation methods can generally be categorized as either being discrepancy based or adversarial based, with the common end goal of minimizing the difference between the source and target distributions. Adversarial methods in particular have become increasingly popular due to their simplicity in training and success in minimizing the domain shift. In this paper we focus on the recently proposed adversarial discriminative domain adaptation (ADDA) (Tzeng et al., 2017), which is related to generative adversarial learning and uses the GAN (Goodfellow et al., 2014) objective to train on the target domain adversarially until it is aligned to the source domain. Whilst ADDA only pretrains the source encoder with source dataset labels, in this paper, we improve on the ADDA framework by first extending the discriminator output over the source classes, in order to additionally incorporate task knowledge into the adversarial loss functions. In adversarial training, we leverage on the fixed distribution over source encoder posteriors, and propose a maximum mean discrepancy (MMD) (Gretton et al., 2012) and reconstruction-based loss function for training a shared encoder and discriminator respectively. We additionally provide an analysis of how our method substantially improves over a base discriminative variant of semi-supervised GANs (Odena, 2016; Salimans et al., 2016). Finally, we evaluate

on standard domain adaptation tasks with digits and Office-31 datasets on which we surpass the performance of ADDA by up to 17% and remain competitive to other recent proposals.

2 Related Work

Discrepancy based methods. Discrepancy based methods typically minimize the maximum mean discrepancy (MMD) (Gretton et al., 2012) loss for this purpose. For example, Tzeng et al. (2014) proposed the deep domain confusion (DDC) method which applied a joint classification and linear MMD loss on an intermediate adaptation layer. Long et al. (2015) extended on DDC by adding multiple task-specific adaptation layers and minimizing the domain shift with a multiple-kernel maximum mean discrepancy. Rather than matching the marginal distributions, the joint adaptation network (JAN) (Long et al., 2016) aligns the domain shift between the joint distributions of input features and output labels. The DSN proposed by Bousmalis et al. (2016) embeds the MMD or adversarial loss as similarity losses in an overarching system of private and shared encoders. Notably, the MMD is commonly used with a Gaussian kernel, which from the Taylor expansion enables matching between all moments of distributions, albeit with some cost in processing. Alternatively, CORAL (Sun & Saenko, 2016) matches only the mean and covariance between distributions whilst still maintaining competitive performance. More recently, Haeusser et al. (2017) proposed associative domain adaptation that replaces the MMD with an efficient discrepancy-based alternative that reinforces association between source and target embeddings.

Adversarial based methods. Adversarial based methods opt for an adversarial loss to minimize the domain shift. The domain adversarial neural network (DANN) (Ganin et al., 2016) first introduced a gradient reversal layer that reversed the gradients of a binary classifier predicting the domain in order to train for domain confusion. Other recent proposals (Liu & Tuzel, 2016; Bousmalis et al., 2017; Taigman et al., 2016) have explored generative models such as GANs (Goodfellow et al., 2014; Mirza & Osindero, 2014) to learn from synthetic source and target data. These approaches typically train two GANs on the source and target input data with tied parameters. In order to circumvent the need to generate images, ADDA (Tzeng et al., 2017) was recently proposed as an adversarial framework for directly minimizing the distance between the source and target encoded representations. A discriminator and target encoder are iteratively optimized in a two-player game akin to the original GAN setting, where the goal of the discriminator is to distinguish the target representation from the source domain and the goal of target encoder is to confuse the discriminator. This implicitly aligns the target distribution to the (fixed) source distribution. The simplicity and power of ADDA has been demonstrated in visual adaptation tasks like MNIST, USPS and SVHN digits datasets. Volpi et al. (2017) further build on ADDA by adding back in a generative component that generates augmented features for more rigorous training.

3 Improving Adversarial Adaptation

We illustrate the framework for improving unsupervised adversarial discriminative domain adaptation in Figure 1. Let $\mathbf{X}_S = \{(\boldsymbol{x}_s^i, y_s^i)\}_{i=0}^{N_s}$ represent the set of source image and label pairs, where $(\boldsymbol{x}_s, y_s) \sim \mathbb{D}_S$, $\mathbf{X}_T = \{(\boldsymbol{x}_t^i)\}_{i=0}^{N_t}$ represent the set of unlabeled target images, $\boldsymbol{x}_t \sim \mathbb{D}_T$ and $\mathbf{X}_B = \hat{\mathbf{X}}_S \cup \mathbf{X}_T$ represent the union of the two sets with $\boldsymbol{x}_b \sim \mathbb{D}_B$, where $\hat{\mathbf{X}}_S \subseteq \mathbf{X}_S$. In the case that $\hat{\mathbf{X}}_S = \emptyset$, then $\mathbb{D}_B = \mathbb{D}_T$ and $\mathbf{X}_B = \mathbf{X}_T$. Let $E_s(\boldsymbol{x}_s; \theta_s)$ represent the source encoder function, parameterized by θ_s which maps an image $\boldsymbol{x}_s$ to the encoder output $\boldsymbol{h}_s$, where $(\boldsymbol{h}_s, y_s) \sim \mathbb{H}_S$. Likewise, let $E_b(\boldsymbol{x}_b; \theta_b)$ represent the shared encoder function, parameterized by θ_b which maps an image $\boldsymbol{x}_b$ to the encoder output $\boldsymbol{h}_b$, where $\boldsymbol{h}_b \sim \mathbb{H}_B$. In addition, C_s represents a classifier function that maps the encoder output $\boldsymbol{h}$ to class probabilities $\boldsymbol{p}$. In this paper, we only consider $\boldsymbol{h}_s$ and $\boldsymbol{h}_b$ as representing the source and shared logits respectively and therefore C_s simply denotes the softmax function on the logits. Finally, let $E_d(\boldsymbol{h}; \phi_d)$ represent an encoder mapping from $\boldsymbol{h}$ to an intermediate representation, and C_d represent a classifier function on said representation; E_d and C_d jointly constitute our discriminator mapping, which we refer to as $D = C_d(E_d)$. Our method consists of three steps, which involve learning the source mapping on the source dataset, adversarial training to align the source and shared domains and finally inferring on the target dataset. The classifier C_s is fully interchangeable between the source encoder E_s and the shared encoder E_b. This means we can embed C_s into the adversarial training of the shared encoder E_b and discriminator D.

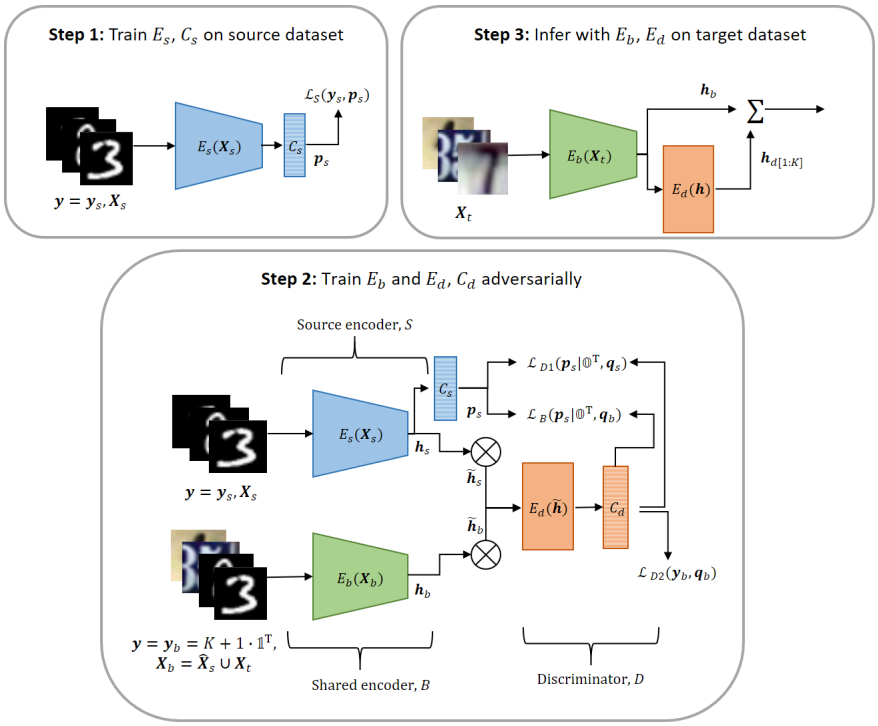


Figure 1: Improved adversarial discriminative domain adaptation. The figure shows the best configuration for training and inference explored in the paper.

3.1 Step 1: Supervised training of the source encoder and classifier

Given that we have access to labels in the source domain, we first train the source encoder E_s and classifier C_s on the source image and label pairs ($\boldsymbol{x}_s$, $y_s \in \{1, ..., K\}$) in a supervised fashion, by minimizing the standard cross entropy loss with K classes:

$$\mathcal{L}_S = -\mathbb{E}_{(\boldsymbol{x}_s, y_s) \sim \mathbb{D}_S} \sum_{k=1}^{K} 1_{[k=y_s]} \log C_s(E_s(\boldsymbol{x}_s)) \tag{1}$$

The source encoder parameters ϕ_s are now frozen, which fixes the distribution $\mathbb{H}_S$. This becomes our reference distribution for adversarial training, analogous to the real image distribution in the GAN setting, where our aim is now to align the shared distribution $\mathbb{H}_B$ to $\mathbb{H}_S$ by learning a suitable shared encoding E_b.

3.2 Step 2: Adversarial training of the shared encoder

3.2.1 Discriminator loss function

We train a shared encoder adversarially by passing the source and shared encoder logits, $\boldsymbol{h}_s$ and $\boldsymbol{h}_b$, to a discriminator D. The shared encoder and discriminator are trained alternately until the discriminator is unable to distinguish between the source and shared domains. In doing so, we implicitly align the shared encoder distribution to that of the source; i.e., $E_b(\boldsymbol{x}_b) \sim \mathbb{H}_S$. As the source encoder has fixed parameters, we learn an asymmetric encoding with untied weights, which is the standard setting in both ADDA (Tzeng et al., 2017) and GAN implementations (Goodfellow et al., 2014; Mirza & Osindero, 2014). In addition, we can improve the convergence properties by first initializing the shared encoder weights with the source encoder weights; i.e., $\theta_t = \theta_s$.

We now consider how to train the shared encoder and discriminator adversarially. Rather than training the discriminator and encoder with the standard GAN loss formulations (i.e., training a logistic

function on the discriminator by assigning labels 0 and 1 to the source and shared domains respectively and training the generator with inverted labels (Goodfellow et al., 2014)), our approach is inspired by semi-supervised GANs (Odena, 2016; Salimans et al., 2016), where it has been found that incorporating task knowledge into the discriminator can jointly improve classification performance and quality of images produced by the generator. Under the discriminative adversarial framework, we can equivalently incorporate task knowledge by replacing the discriminator logistic function with a $K+1$ multi-class classifier, where C_d simply denotes the softmax function on the discriminator logits. As such, the discriminator output $\boldsymbol{q}$ is a $K+1$ dimensional vector representing the class probabilities, in which the first K dimensions represent the task-specific classes $y \in \{1, \ldots, K\}$ and the final $K+1$ dimension represents the 'shared' class $y = K+1$, assigned to inputs from the shared encoder. However, contrary to semi-supervised GANs, the discriminator inputs and outputs now share common supports over the K task classes. For the source domain, we can leverage on this fact by effectively modelling the discriminator as a denoising autoencoder (Vincent et al., 2008), where we can jointly train the discriminator to reconstruct the source encoder logits and encourage the discriminator to learn something more informative by corrupting its inputs. We refer to the corruption process as $N(\tilde{\boldsymbol{h}}_s|\boldsymbol{h}_s)$, which represents the conditional distribution over the corrupted source encoder logits $\tilde{\boldsymbol{h}}_s$ given the source encoder logits $\boldsymbol{h}_s$. Therefore, the first term of our discriminator loss function is effectively a reconstruction loss, which we set as the cross entropy between the transformed source encoder posteriors $\hat{\boldsymbol{p}}_s = C_s(\boldsymbol{h}_s/T)||0$ and source discriminator posteriors $\boldsymbol{q}_s$ (i.e., post-softmax), where $||$ denotes a concatentation operation and T is a temperature constant:

$$\begin{aligned}\mathcal{L}_{D1} &= -\mathbb{E}_{(\boldsymbol{h}_s, y_s)\sim\mathbb{H}_S}\mathbb{E}_{\tilde{\boldsymbol{h}}_s\sim N(\tilde{\boldsymbol{h}}_s|\boldsymbol{h}_s)}(C_s(\boldsymbol{h}_s/T)||0 \cdot \log\left(D(\tilde{\boldsymbol{h}}_s)\right)) \\ &= -\mathbb{E}_{(\boldsymbol{h}_s, y_s)\sim\mathbb{H}_S}\mathbb{E}_{\tilde{\boldsymbol{h}}_s\sim N(\tilde{\boldsymbol{h}}_s|\boldsymbol{h}_s)}\sum_{k=1}^{K}\hat{\boldsymbol{p}}_{s,k}\log(\boldsymbol{q}_{s,k})\end{aligned} \tag{2}$$

Notably, we append a zero to the source encoder posteriors to represent the $K+1$-th 'shared' class, which maintains a valid probability distribution (sums to 1), whilst enforcing a zero probability that the posteriors were generated by the shared encoder. We additionally soften the source encoder posterior distribution by dividing the source encoder logits $\boldsymbol{h}_s$ by temperature T, in order to further deviate from the discriminator learning an identity function. In this paper, the corruption process N is simply configured as dropout on the encoder logits; it is worth noting that a keep probability z greater than 0.5 generally maintains overlapping class supports between the encoder and discriminator posteriors.

We also apply dropout independently to the shared encoder logits $\boldsymbol{h}_b$, in order to symmetrize the source and shared encoder inputs presented to the discriminator. However, we want the discriminator to distinguish between the source and shared encoder logits. We train the discriminator to assign the $K+1$-th class to the corrupted shared encoder logits $\tilde{\boldsymbol{h}}_s$, such that they lie in an orthogonal space to the source domain. In other words, the second term of our discriminator loss function for the shared encoder logits is:

$$\mathcal{L}_{D2} = -\mathbb{E}_{(\boldsymbol{h}_b, y_b)\sim\mathbb{H}_B}\mathbb{E}_{\tilde{\boldsymbol{h}}_b\sim N(\tilde{\boldsymbol{h}}_b|\boldsymbol{h}_b)}\sum_{k=1}^{K+1}1_{[k=K+1]}\log(D(\tilde{\boldsymbol{h}}_b)) \tag{3}$$

The discriminator loss function $\mathcal{L}_D$ is thus simply the sum of (2) and (3): $\mathcal{L}_D = \mathcal{L}_{D1} + \mathcal{L}_{D2}$.

3.2.2 SHARED ENCODER LOSS FUNCTION

In order to train the shared encoder adversarially, we want the shared encoder to generate an output that is representative of one of the first K task-specific classes rather than the $K+1$-th 'shared' class that it is assigned when training the discriminator. To achieve this, we leverage on the two source posteriors, $\boldsymbol{p}_s$ and $\boldsymbol{q}_s$, generated by the source encoder and discriminator respectively. Contrary to supervised domain adaptation methods, there are no known source and shared pairwise correspondences and we cannot formulate a paired test over the posteriors. However, we can formulate the problem as a two-sample test by considering the distribution over shared discriminator posteriors $\boldsymbol{q}_b$ compared to the distribution over the source encoder posteriors, where our null hypothesis is that

the distributions are equal. We consider a set of shared posteriors $\mathbf{Q}_B = \{\boldsymbol{q}_b^1, \ldots, \boldsymbol{q}_b^m\} \sim \mathbb{Q}_B$ and a set of source posteriors $\mathbf{P}_S = \{\boldsymbol{p}_s^1, \ldots, \boldsymbol{p}_s^n\} \sim \mathbb{P}_S$. Effectively, we want to minimize the distance between $\mathbb{P}_S$ and $\mathbb{Q}_B$ without performing any density estimation. To this end, we adopt the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) metric as a measure of distance between the mean embeddings of $\boldsymbol{p}_s$ and $\boldsymbol{q}_b$. For reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, function class $\mathcal{F}$ = $\{f : \|f\| \leq 1\}$ and infinite dimensional feature map $\phi : \mathcal{X} \rightarrow \mathcal{H}$ the MMD can be expressed as:

$$\mathcal{D}_{\text{MMD}} = \sup_{f \in \mathcal{F}, \|f\|_{\mathcal{H}} \leq 1} |\mathbb{E}_{\boldsymbol{p}_s \sim \mathbb{P}_S} f(\boldsymbol{p}_s||0) - \mathbb{E}_{\boldsymbol{q}_b \sim \mathbb{Q}_B} f(\boldsymbol{q}_b)| = \|\mathbb{E}_{\boldsymbol{p}_s \sim \mathbb{P}_S} \phi(\boldsymbol{p}_s||0) - \mathbb{E}_{\boldsymbol{q}_b \sim \mathbb{Q}_B} \phi(\boldsymbol{q}_b)\|_{\mathcal{H}} \tag{4}$$

We again append a 0 to the source encoder posteriors to represent the shared class probability, such that both source and target posteriors are $K + 1$ dimensional prior to mapping to $\mathcal{H}$. This zero constraint on the $K + 1$-th class acts as a stronger prior upon which to learn the shared encoder; as such, the source encoder posterior provides a more informative representation than the source discriminator posterior. It is additionally worth noting that MMD employed in our proposal can be interpreted as matching all moments between the source and shared posterior distributions, whereas conventional feature matching (as in Salimans et al. (2016)) is only empirically matching the first order moments (means) of the intermediate discriminator layer activations. The feature map ϕ in (4) corresponds to a PSD kernel k such that $k(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle_{\mathcal{H}}$, which means we can rewrite (4) in terms of k. The loss function on our shared encoder that we wish to minimize can thus be written as:

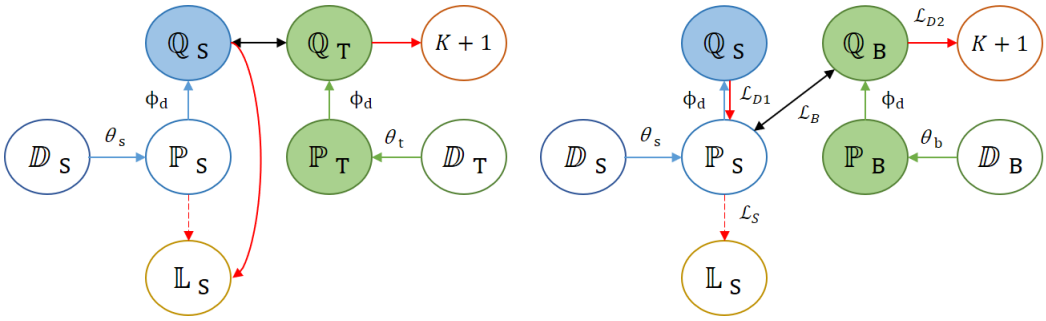
$$\mathcal{L}_B = {\mathcal{D}_{\text{MMD}}}^2 = \mathbb{E}_{\boldsymbol{p}_s, \boldsymbol{p}_s' \sim \mathbb{P}_S, \mathbb{P}_S} k(\boldsymbol{p}_s||0, \boldsymbol{p}_s'||0) - \mathbb{E}_{\boldsymbol{p}_s, \boldsymbol{q}_b \sim \mathbb{P}_S, \mathbb{Q}_B} k(\boldsymbol{p}_s||0, \boldsymbol{q}_b) + \mathbb{E}_{\boldsymbol{q}_b, \boldsymbol{q}_b' \sim \mathbb{Q}_B, \mathbb{Q}_B} k(\boldsymbol{q}_b, \boldsymbol{q}_b') \tag{5}$$

In this paper we opt to use a linear combination of r multiple RBF kernels over a range of standard deviations, such that $k(\boldsymbol{x}, \boldsymbol{y}) = \sum_r \exp\{-\frac{1}{2\sigma_r} \|\boldsymbol{x} - \boldsymbol{y}\|^2\}$, where σ_r is the standard deviation of the r-th RBF kernel. We find that the standard RBF kernel as above performs better in practice than a generalized RBF kernel with a distribution based metric such as chi-squared distance or squared Hellinger's distance, although these are viable options. By introducing a linear combination over varying bandwidths, we improve the generalization performance over different sample distributions. This method of generalization with fixed kernels is commonly used both in generative models (Li et al., 2015; Dziugaite et al., 2015) and other domain adaptation discrepancy based methods (Bousmalis et al., 2016; Long et al., 2015). To ensure consistency, we fix the kernel combination for all experiments. Specifically, after experimentation, we found that optimal performance for our framework is achieved with a summation over five kernels, with $\sigma_r = 10^{-r}, r \in \{0, \ldots, 4\}$. Finally, we note that, in order to improve the generalization in the context of GANs, recent work (Li et al., 2017; Mroueh et al., 2017) has looked at kernel optimization by adversarially training the kernel with the discriminator or critic, such that the kernel is maximally discriminative. While, we can also perform this kernel optimization in the discriminative adversarial setting with multiple classes, we leave this for future work.

3.3 Step 3: Inference on the target dataset

After training the shared encoder, we can now perform inference on the target dataset. However, we have effectively trained two sets of logits on target examples; namely, the mapped shared encoder output $\boldsymbol{h}_b$ and the discriminator output $\boldsymbol{h}_d$. In the optimal setting, where we have trained the discriminator to equilibrium, we would expect the discriminator mapped source and shared distributions would be aligned - however, in practice we tend to gain 1-2% on test datasets by averaging the K task components of the encoder and discriminator logits. Our final label prediction $\hat{y}$ is computed as:

$$\hat{y} = \arg \max_{j \in 1, \ldots, K} (\boldsymbol{h}_b + \boldsymbol{h}_{d[1:K]}) \tag{6}$$



(a) Base (b) Improved

Figure 2: (Best viewed in color) Computational graphs for a) our base discriminative variant to semi-supervised GANs and b) our improved model for discriminative adversarial training. Each node is represented with its modelled distribution. Blue nodes/arrows represents the source domain components and green nodes/arrows represent the target/shared domain components. Nodes that are not colored in are fixed during adversarial training and those that are colored in are trainable. Red and black arrows represent the discriminator and encoder loss components respectively and their arrow direction represents the direction of alignment for asymmetric losses.

3.4 Comparisons with discriminative variants to semi-supervised GANs

In order to substantiate the novelty in our proposal, Figure 2 presents the computational graphs during adversarial training for (a) a base discriminative variant to semi-supervised GANs and (b) our improved model for discriminative adversarial training. Each node is displayed with the corresponding distribution being modelled, with $\mathbb{L}_S$ representing the source task label distribution and $K+1$ representing the fixed encoder label. The base method in (a) trains an encoder on target examples from $\mathbb{D}_T$ only and learns the encoder and discriminator posterior distributions $\mathbb{P}_T$, $\mathbb{Q}_S$ and $\mathbb{Q}_T$ respectively during adversarial training. The discriminator is trained by aligning the source and target discriminator posteriors to their respective labels with cross entropy loss and the encoder is trained to minimize a discrepancy-based loss between the source and target disciminator posteriors (or intermediate discriminator layers). In our experiments, we adopt maximum mean discrepancy for this loss, with kernels configured as in Section 3.2.2.

On the other hand, our proposal in (b) trains a shared encoder on target and a subset of source examples from $\mathbb{D}_B$, where the additional source examples ensure that $\mathbb{P}_S$ and $\mathbb{P}_B$ (and $\mathbb{Q}_S$ and $\mathbb{Q}_B$) have overlapping supports, in a similar manner to label smoothing (Salimans et al., 2016), except the mapped source examples are now parameterized by θ_b. Importantly, we recognize that the distribution of the source encoder posteriors $\mathbb{P}_S$ is fixed and only changes stochastically with mini-batch; as such, we centralize our discriminator and encoder loss functions around this distribution. This, along with the hard zero constraint on the source encoder posteriors for the $K+1$-th class probability, is key for stabilizing training of the shared encoder. For the discriminator loss $\mathcal{L}_D$, aligning the source discriminator to the softened source encoder posteriors in our proposal enables the discriminator to quickly learn inter-relationships between classes. For the encoder loss $\mathcal{L}_B$, $\mathbb{Q}_B$ is aligned to the fixed $\mathbb{P}_S$, which is favorable to the base method where both target $\mathbb{Q}_T$ and the reference $\mathbb{Q}_S$ are changing with time. We note that we can provide an additional constraint by maximizing a discrepancy loss between $\mathbb{Q}_S$ and $\mathbb{Q}_B$ when training the discriminator but we found that, in practice, this did not improve results in our tests.

4 Experimental Results

We present experimental results on the unsupervised domain adaptation task. In order to compare with ADDA and other recently proposed methods, we experiment on four digits datasets of varying sizes and difficulty: MNIST-M (Ganin et al., 2016), MNIST (LeCun et al., 1998), USPS and SVHN (Netzer et al., 2011). We demonstrate substantial gain over ADDA and other recent methods, which is evident on the more difficult domain adaptation tasks such as SVHN → MNIST. We additionally

Method	SVHN → MNIST	USPS → MNIST	MNIST → USPS	MNIST → MNIST-M
Source only	0.644	0.597	0.754	0.705
DANN Ganin et al. (2016)	0.739	0.730	0.771	0.529
DDC Tzeng et al. (2014)	0.681	0.665	0.791	-
DSN Bousmalis et al. (2016)	0.827	-	-	0.832
DTN Taigman et al. (2016)	0.844*	-	-	-
UNIT Liu et al. (2017)	0.905*	-	-	-
CoGAN Liu & Tuzel (2016)	no convergence	0.891	0.912	-
RAAN Chen et al. (2018)	0.892	0.921	0.890	**0.985**
ADDA Tzeng et al. (2017)	0.760 (26%)	0.901 (58%)	0.894 (19%)	0.800 (14%)**
DIFA Volpi et al. (2017)	0.897 (32%)	0.897 (43%)	**0.923 (28%)**	-
Base (Fig. 2(a))	0.767 (19%)	0.914 (53%)	0.857 (14%)	0.921(31%)
Improved 1 (target only, $z = 1.0$)	0.863 (34%)	0.925 (55%)	0.854 (13%)	0.930 (32%)
Improved 2 (source + target, $z = 1.0$)	0.899 (40%)	0.939 (57%)	0.907 (20%)	0.920 (31%)
Improved 3 (source + target, $z = 0.7$)	**0.927 (44%)**	**0.948 (59%)**	0.910 (21%)	0.915 (30%)

Table 1: Accuracy for our base configuration (Figure 2(a)) and 3 variants of our proposed method (Figure 2(b)) compared to the current state-of-the-art. 'Target only' and 'source + target' refer respectively to the shared encoder being trained on target examples only or both source and target examples. In order to isolate the performance gain from domain adaptation for our proposals, we report in parentheses the percentage increase (relative) over the source-only accuracy, as reported in the respective papers for DIFA (Volpi et al., 2017) and ADDA (Tzeng et al., 2017).*UNIT (Liu et al., 2017) and DTN (Taigman et al., 2016) use additional SVHN data (131 images and 531 images respectively). **This is our implementation of ADDA (Tzeng et al., 2017) on MNIST → MNIST-M, as this task is not used in the original paper.

report accuracy on the Office-31 dataset (Saenko et al., 2010) compared to the current state-of-the-art methods.

4.1 DIGITS DATASETS

We consider four standard domain adaptation scenarios between dataset pairs drawn from MNIST-M (Ganin et al., 2016), MNIST (LeCun et al., 1998), USPS and SVHN (Netzer et al., 2011) digits datasets, which are each comprised of $K = 10$ digit classes (0-9). Specifically, we evaluate on MNIST → USPS, USPS → MNIST, SVHN → MNIST and MNIST → MNIST-M. The difficulty in domain adaptation task increases as the variability between datasets increases. We follow a similar training procedure of Tzeng et al. (2017). For the MNIST → USPS and USPS → MNIST experiments, we sample 2000 images from MNIST and 1800 from USPS, otherwise we train and infer on the full datasets. For MNIST → MNIST-M, we generate the unlabelled MNIST-M target dataset by following the process described by Ganin *et al.* (Ganin et al., 2016). For all experiments we use a modified LeNet architecture (LeCun et al., 1998) for the source and target encoder. The discriminator is comprised of 2 fully connected layers with 500 hidden units and a final fully-connected layer with $K + 1 = 11$ hidden units that outputs the logits. With this setup, our network is roughly the same complexity as ADDA in terms of number of parameters. In step 1, the source encoder is trained with the Adam (Kingma & Ba, 2014) optimizer for 10k iterations with a batch size of 128 and learning rate of 0.001. In step 2, the target/shared encoder is trained with a batch size of 256 per domain for 10k iterations but with a lower learning rate of 0.0002, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We set the temperature constant T in (2) to 2.0 for all experiments. We resize all images to a fixed size of 28×28 prior to CNN processing. Additonally, we use data augmentation for MNIST → MNIST-M by randomly inverting the MNIST grayscale values and replicating the MNIST inputs channel-wise to match MNIST-M dimensions. Our results are provided in Table 1 compared to the current state-of-the-art and when training on source only. We focus our comparison on ADDA (Tzeng et al., 2017) and DIFA (Volpi et al., 2017), which are recently proposed adversarial methods.

We report accuracy for the base configuration of Figure 2(a) plus variants of our proposal of Figure 2(b). We vary our proposal by training a shared encoder either with only target inputs drawn from the target distribution only $\mathbb{D}_T$ or with source and target inputs from $\mathbb{D}_B$, and varying the level of corruption N in the discriminator loss via the dropout keep probability z. We denote each variant as improved proposal 1, 2 and 3. In order to isolate the performance gain from domain adaptation, we compute the percentage increase (relative) over the source only accuracy reported in the paper

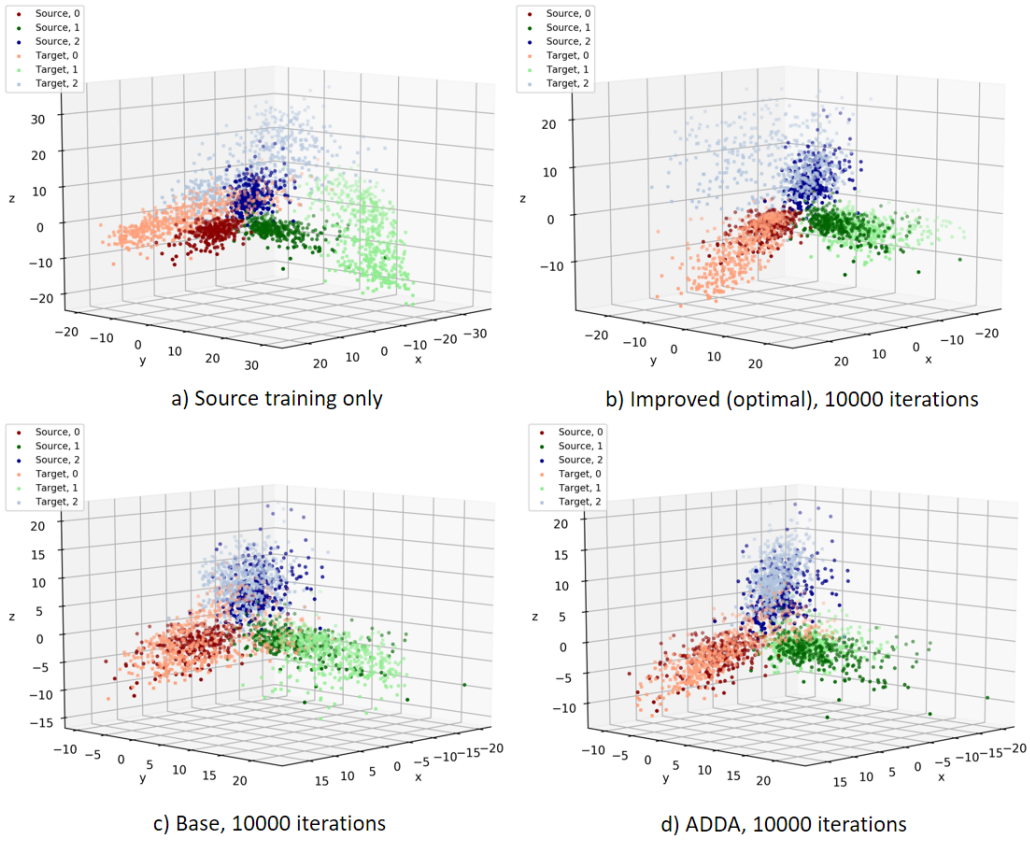


Figure 3: (Best viewed in color) 3D scatter plots for a subset of source and target logits for the SVHN → MNIST domain adaptation task on 3 classes only (0, 1 and 2). Source and target examples are randomly selected from the SVHN and MNIST test datasets respectively for visualization.

(shown in parentheses in Table 1). First, we note that switching the loss functions from the base configuration to proposal 1, which uses the target encoder and no corruption, there is a substantial increase in accuracy on SVHN → MNIST. Switching from the target to shared encoder in proposal 2 provides further accuracy gain and supplementing this with dropout in proposal 3 ($z = 0.7$), in order to corrupt the source and shared encoder logits, gives optimal performance on the majority of datasets. On average over all datasets, our proposals outperform DIFA, RAAN and ADDA. We note that on MNIST → MNIST-M, the base method and all proposals perform very similarly; this is attributed to the very low variance of the source distribution compared to the target, such that regularizing the encoder with source examples in the shared encoder has minimal effect.

Rather than using a reduction method such as t-SNE (Maaten & Hinton, 2008) that introduces additional hyperparameters such as perplexity to visualize the domain shift, we instead present 3D scatter plots in Figure 3 of the source and target logits when trained on 3 classes only from the SVHN → MNIST domain adaptation task, in order to further validate the performance of our method. For (b)-(d), adversarial training is stopped after 10000 iterations. As is evident from the Figure, whilst ADDA is able to learn a better approximation to the source distribution, it is unable to learn class separation around the origin, where the logit distribution is more uniform. This is also apparent in the base configuration and as such, both methods misclassify a sizeable proportion of zero examples, achieving an overall accuracy of around 85% on the test dataset. On the other hand, our proposal forgoes a tight approximation to the source for better class separation and achieves an accuracy of 98%.

Method	A → W	A → D	D → A
Source only	0.707	0.720	0.581
DANN Ganin et al. (2016)	0.730	0.723	0.534
DDC Tzeng et al. (2014)	0.618	0.644	0.521
DRCN Ghifary et al. (2016)	0.687	0.668	0.560
JAN Long et al. (2016)	0.752	0.728	0.575
ADDA Tzeng et al. (2017)	0.751	-	-
Improved (target only, $z = 0.7$)	**0.821**	0.799	0.610
Improved (source + target, $z = 0.7$)	0.798	**0.807**	**0.639**

Table 2: Accuracy for improved (Figure 2(b)) configurations compared to state-of-the-art on the Office-31 dataset. 'Target only' and 'source + target' refer respectively to the shared encoder being trained on target examples only or both source and target examples.

4.2 OFFICE-31 DATASET

We report results on the standard Office-31 (Saenko et al., 2010) dataset in the fully transductive setting. The Office-31 dataset consists of 4,110 images spread across 31 classes in 3 domains: Amazon, Webcam, and DSLR. Our results focus on the three of the more difficult domain adaptation tasks; Amazon → Webcam (A → W), Amazon → DSLR (A → D) and DSLR → Amazon (D → A). In order to demonstrate the strength of our proposal, we use VGG-16 pre-trained on ImageNet and fine-tune only the final fully-connected layer. We train with stochastic gradient descent and a learning rate of 0.001 and temperature constant T set to 2.0. Our discriminator is restricted to only 500 hidden units per layer and we only train adversarially for 2k iterations. We note that the number of training parameters is 377 thousand in total, compared to over 6 million utilized for ADDA (Tzeng et al., 2017). Despite only training on a small subset of total parameters, both improved variants remain competitive or surpass the performance of other recent methods. We additionally note that under our training setup, ADDA consistently obtains a degenerate solution due to instability during training.

5 CONCLUSION

We extend adversarial discriminative domain adaptation by explicitly accounting for task knowledge in the discriminator during adversarial training and leveraging on the fixed distribution over source encoder posteriors, with which we derive our adversarial loss function. In particular, we consider the discriminator as a denoising autoencoder in its corresponding loss function and minimize the maximum mean discrepancy between the discriminator posterior and source encoder posterior distribution to train the encoder. We additionally compare our approach with a base discriminative variant of semi-supervised GANs. Our framework is shown to compete or outperform the state-of-the-art in unsupervised transfer learning on standard datasets, while remaining simple and intuitive to use and can be extended further in future work by embedding kernel optimization into the adversarial framework.

REFERENCES

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pp. 343–351, 2016.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 7, 2017.

Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7976–7985, 2018.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pp. 597–613. Springer, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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

Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *International Conference on Computer Vision (ICCV)*, volume 2, pp. 6, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2200–2210, 2017.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727, 2015.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pp. 469–477, 2016.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pp. 700–708, 2017.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. *arXiv preprint arXiv:1605.06636*, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Youssef Mroueh, Tom Sercu, and Vaibhava Goel. Mcgan: Mean and covariance feature matching gan. *arXiv preprint arXiv:1702.08398*, 2017.

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

Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pp. 443–450. Springer, 2016.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 4, 2017.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.

Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. *arXiv preprint arXiv:1711.08561*, 2017.