Glocal Alignment for Unsupervised Domain Adaptation

Sachin Chhabra
schhabr6@asu.edu
Arizona State University
Tempe, AZ, USA

Prabal Bijoy Dutta
pdutta6@asu.edu
Arizona State University
Tempe, AZ, USA

Baoxin Li
baoxin.li@asu.edu
Arizona State University
Tempe, AZ, USA

Hemanth Venkateswara
hemanthv@asu.edu
Arizona State University
Tempe, AZ, USA

ABSTRACT

Traditional unsupervised domain adaptation methods attempt to align source and target domains globally and are agnostic to the categories of the data points. This results in an inaccurate categorical alignment and diminishes the classification performance on the target domain. In this paper, we alter existing adversarial domain alignment methods to adhere to category alignment by imputing category information. We partition the samples based on category using source labels and target pseudo labels and then apply domain alignment for every category. Our proposed modification provides a boost in performance even with a modest pseudo label estimator. We evaluate our approach on 4 popular domain alignment loss functions using object recognition and digit datasets.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; *Image representations.*

KEYWORDS

Unsupervised domain adaptation; Local alignment; Domain alignment; Category alignment

ACM Reference Format:
Sachin Chhabra, Prabal Bijoy Dutta, Baoxin Li, and Hemanth Venkateswara. 2021. Glocal Alignment for Unsupervised Domain Adaptation. In *Proceedings of the 1st Workshop on Multimedia Understanding with Less Labeling (MULL '21), October 24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3476098.3485051

1 INTRODUCTION

In real-world applications, train and test data generally do not belong to the same distribution. Even though the classification model is well-trained on the training dataset (also known as the source), this change in distribution often yields to the degraded performance because of the covariance shift [22]. The standard solution is to fine-tune the network on the new domain (also known as the target) but it requires a significant amount of labeled data. This may not be a viable option as labeling the data is an expensive task. To solve this problem, Unsupervised Domain Adaptation techniques aim to learn target labels by using labeled source and unlabelled target data only [19].

Unsupervised domain adaptation problem has been researched extensively over the past decade. The standard solution is to classify samples based on features that are invariant between the source and target domains. To attain this objective, most techniques, attempt to reduce the distance between the generated features of the domains using a distance-metric like maximum mean discrepancy [10], Wasserstein distance [17] or adversarially using a discriminator [4, 20, 21]. These methods make the features domain-invariant but the alignment is done without taking category information of the features into consideration and often leads to jumbled classes.

To improve adversarial global domain alignment, we propose to align the data points locally using category information - à la, *Think Globally, Act Locally.* First, we partition the samples of the

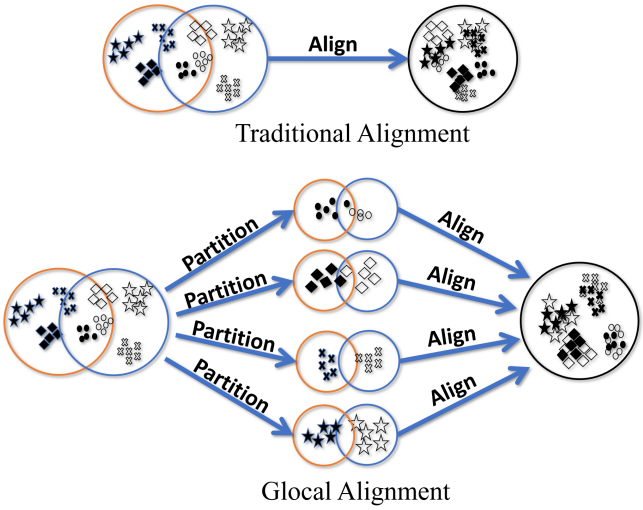

Figure 1: Illustration of the Glocal Method. We identify pseudo labels to partition the categories and align the domains locally which results in an effective global domain alignment.

MULL '21, October 24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8681-4/21/10…$15.00
https://doi.org/10.1145/3476098.3485051

Method	Discriminator Loss	Generator Loss
DANN	$-\mathbb{E}_{x\sim D_s}[\ln D(G(x))] - \mathbb{E}_{x\sim D_t}[\ln(1-D(G(x)))]$	Gradient Reversal
MDC	$-\mathbb{E}_{x\sim D_s}[\ln D(G(x))] - \mathbb{E}_{x\sim D_t}[\ln(1-D(G(x)))]$	$-\mathbb{E}_{x\sim (D_s\cup D_t)}[\frac{1}{2}\ln D(G(x)) + \frac{1}{2}\ln(1-D(G(x)))]$
GAN_1	$-\mathbb{E}_{x\sim D_s}[\ln D(G(x))] - \mathbb{E}_{x\sim D_t}[\ln(1-D(G(x)))]$	$-\mathbb{E}_{x\sim D_t}[\ln D(G(x))]$
GAN_2	$-\mathbb{E}_{x\sim D_s}[\ln D(G(x))] - \mathbb{E}_{x\sim D_t}[\ln(1-D(G(x)))]$	$-\mathbb{E}_{x\sim D_s}[\ln(1-D(G(x)))] - \mathbb{E}_{x\sim D_t}[\ln D(G(x))]$

Table 1: Traditional Domain Alignment loss functions.

Method	Discriminator Loss	Generator Loss
$\mathcal{G}$-*DANN*	$-\mathbb{E}_{x\sim D_s}[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))]$ $-\mathbb{E}_{x\sim D_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln(1-D^i(G(x)))]$	Gradient Reversal using $-\mathbb{E}_{x\sim D_s}[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))]$ $-\mathbb{E}_{x\sim D'_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln(1-D^i(G(x)))]$
$\mathcal{G}$-*MDC*	$-\mathbb{E}_{x\sim D_s}[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))]$ $-\mathbb{E}_{x\sim D_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln(1-D^i(G(x)))]$	$-\mathbb{E}_{x\sim (D_s\cup D'_t)}[\frac{1}{2}\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))$ $+\frac{1}{2}\sum_{i=1}^{K}\mathbb{1}(i=y)\ln(1-D^i(G(x)))]$
$\mathcal{G}$-GAN_1	$-\mathbb{E}_{x\sim D_s}[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))]$ $-\mathbb{E}_{x\sim D_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln(1-D^i(G(x)))]$	$-\mathbb{E}_{x\sim D'_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln D^i(G(x))]$
$\mathcal{G}$-GAN_2	$-\mathbb{E}_{x\sim D_s}[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))]$ $-\mathbb{E}_{x\sim D_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln(1-D^i(G(x)))]$	$-\mathbb{E}_{x\sim D_s}[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln(1-D^i(G(x)))]$ $-\mathbb{E}_{x\sim D'_t}[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln D^i(G(x))]$

Table 2: Glocal Domain Alignment loss functions.

source and target that belong to the same classes. In the case of unsupervised domain adaptation, since the target labels are not available, we use the pseudo labels instead. Pseudo-Label is a semi-supervised learning technique that treats network prediction as a true label if the probability of a predicted category is above a predetermined threshold [9]. We then align datapoints from each category for both the domains. The local category alignment, in turn ensures global domain alignment. This simple modification can be applied to any adversarial domain alignment techniques and results in a better alignment. Our approach is depicted in Fig. 1. In this paper, we experiment on DANN [4], MDC [20], GAN loss functions [5]. We compare these losses as well as our modification to them on a common network. These loss functions are discussed in section 3. In the next section, we discuss the related work. Section 4 outlines the Glocal alignment method. Section 5 discusses experiments and results followed by conclusion in Section 6.

2 RELATED WORK

Domain adaptation techniques aim to reduce the distribution discrepancy between source and target domains using a domain alignment loss. We discuss four adversarial based alignment techniques in the Background section. These alignment losses can be combined [12] or applied at multiple layers [10, 12] to further boost performance. State-of-the-art domain alignment techniques use additional loss functions for alignment along with the adversarial domain alignment loss. Soft labels generated using source samples were used to train the target in [20]. Maximum Mean Distance was reduced between source and target in [12]. A combination of Lipschitz constraint and low entropy based loss function was used in [18].

Recent approaches have emphasized the problems of mere global adaptation. A moving class-wise centroid based distance was implemented in [25]. The class-wise centroid distance was reduced progressively in [2]. The discriminator can also be conditioned on the class for encouraging class-wise features distinction [11]. An image-to-image translation based technique ensured a generated target image has the same class label as the original source image [6]. We promote the local alignment and propose to modify the existing global adversarial alignment loss functions to operate locally.

3 BACKGROUND

Let $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ be the source domain consisting of n_s labeled images sampled from distribution S. Likewise the target domain is denoted as $D_t = \{(x_i^t)\}_{i=1}^{n_t}$ consisting of n_t unlabeled images sampled from distribution T. The goal of unsupervised domain adaptation is to learn the target labels $\{y_t^i\}_{i=1}^{n_t}$ using D_s and D_t. It is assumed that D_s and D_t have an identical label space of K categories but since $S \neq T$, a classifier trained using D_s will under perform when trying to predict the target data labels.

We propose to align the source and target distributions using adversarial feature alignment based on the Generative Adversarial Network (GAN) [5]. The standard GAN model consists of a Generator network $G(.)$, and a Discriminator network $D(.)$ which are pitted against each other in min-max optimization. Traditionally, generator takes noise as an input and outputs an image but in the case of unsupervised domain adaptation, the generator acts as a feature extractor/encoder. It takes image as the input and outputs deep features which can be classified by a classification module to appropriate classes. The Discriminator network attempts to distinguish between the source and target features and the Generator attempts to align the features so that they are indistinguishable by the discriminator. Upon convergence, the marginal distributions of the source and target are said to be aligned. In the following we outline 4 popular variants of adversarial alignment.

3.1 Vanilla GAN

The first model GAN_1 is the vanilla GAN model where the Generator $G(.)$ attempts to align the source and target features and the Discriminator $D(.)$ learns to distinguish between them. The system also has a classifier $C(.)$ to classify the aligned features. The objective function for this model is,

$$\begin{aligned}&\min_{D} -\mathbb{E}_{x\sim D_s}[\ln D(G(x))] - \mathbb{E}_{x\sim D_t}[\ln(1-D(G(x)))]\\&\min_{G,C} -\lambda_{g_1}\,\mathbb{E}_{x\sim D_t}[\ln D(G(x))] + \mathcal{L}^s_{ce},\end{aligned} \tag{1}$$

where $\mathcal{L}_{ce} = -\mathbb{E}_{(x,y)\sim D_s}[y\cdot \ln C(G(x))]$ is the cross entropy loss and λ_{g_1} models the importance of the alignment loss. In Eq. 1, the Generator attempts to align target features to a fixed source distribution. Alternatively, we get more leverage in the alignment space when modify both the source and target features [18]. While the objective function for training $D(.)$ is the same as in Eq. 1, the objective function to train the Generator $G(.)$ and Classifier $C(.)$ is,

$$\min_{G,C} -\lambda_{g_2}\big\{\mathbb{E}_{x\sim D_s}[\ln(1-D(G(x)))] + \mathbb{E}_{x\sim D_t}[\ln D(G(x))]\big\} + \mathcal{L}^s_{ce} \tag{2}$$

We refer to Eq. 1 as GAN_1 and to Eq. 2 as GAN_2.

3.2 DANN

Our third model is the popular Domain-Adversarial Training of Neural Networks (DANN) [4]. In DANN the Generator $G(.)$ uses a reversed gradient to update parameters during training in order to confuse the Discriminator $D(.)$. The objective function for the Generator differs from Eq. 1 as,

$$\min_{G,C} -\lambda_d\big\{\mathbb{E}_{x\sim D_s}[\ln D(G(x))] + \mathbb{E}_{x\sim D_t}[\ln(1-D(G(x)))]\big\} + \mathcal{L}^s_{ce}. \tag{3}$$

3.3 MDC

The 4th model for comparison is the Maximum Domain Confusion (MDC) [20]. GAN loss is the standard choice for training a network to mimic a distribution. The MDC introduces maximum domain confusion through a cross entropy loss between the output of the discriminator and the uniform distribution. This results in even the best discriminator performing poorly, thereby aligning the domains. The loss function for the Discriminator is identical to the one in Eq. 1. The objective function for the Generator and Classifier is given by,

$$\min_{G,C} -\lambda_m\big\{\mathbb{E}_{x\sim(D_s\cup D_t)}\big[\frac{1}{2}\ln D(G(x)) + \frac{1}{2}\ln(1-D(G(x)))\big]\big\} + \mathcal{L}^s_{ce}. \tag{4}$$

The summary of these loss functions is presented in Table 1.

4 GLOCAL METHOD

Traditional domain alignment aligns the deep features without any regards to their category. Such alignment generally results in aligning mismatched classes and hurts target performance. To overcome this issue, we propose Glocal alignment that splits data into classes and performs category-level (local) adversarial alignment. Since in case of unsupervised domain adaptation we do not have target labels, we rely on the pseudo label generated by the network. This way Glocal alignment achieves global alignment by aligning data points from each category for the two domains.

We train the classifier on source dataset D_s using standard cross-entropy loss $\mathcal{L}^s_{ce}$. Next, we determine the pseudo-labels for the target data by applying a threshold on the classifier prediction,

$$\hat{y}^t_i = \begin{cases}\underset{y}{\text{argmax}}\; p(y|C(x^t_i)), & \text{if } \max p(y|C(x^t_i)) > \tau\\ -1 & \text{otherwise.}\end{cases} \tag{5}$$

We define $D'_t := \{(x^t_i, \hat{y}^t_i)\}_{i=1}^{n'_t} | \hat{y}^t_i \neq -1\}$ as the pseudo-labeled target dataset. To apply alignment on each class, we would need K discriminators and the amount of data available to train each discriminator will also reduce by a factor of K, making this approach impractical when K is large. To address this concern, we use multi-task learning approach and modify the discriminator to multi-headed logit [15]. Specifically, we change the number of outputs of the discriminator from 1 (Global) to K and each output head acts as a decision function for one of K categories. D^i represents the sigmoid output of the multi-headed discriminator at the i^{th} head. The glocal discriminator loss is defined as,

$$\begin{aligned}&\min_{D} -\mathbb{E}_{x\sim D_s}\big[\sum_{i=1}^{K}\mathbb{1}(i=y)\ln D^i(G(x))\big]\\&-\mathbb{E}_{x\sim D_t}\big[\sum_{i=1}^{K}\mathbb{1}(i=\hat{y}^t)\ln(1-D^i(G(x)))\big]\end{aligned} \tag{6}$$

Another issue that arises is the class-imbalance problem due to use of a subset of the target samples only. The reduced number of target samples introduces a bias in the discriminator towards the source domain. To overcome this issue, we do not use the threshold τ while training the discriminator. The discriminator heads are trained using all the target samples D_t based on their pseudo labels, enabling the discriminator D to learn the actual distributions without any bias. Only confident samples D'_t are used for training the feature extractor G which are aligned with the source. Similar to the discriminator, we modify the generator loss functions by adding the condition to select the appropriate discriminator head and use D'_t

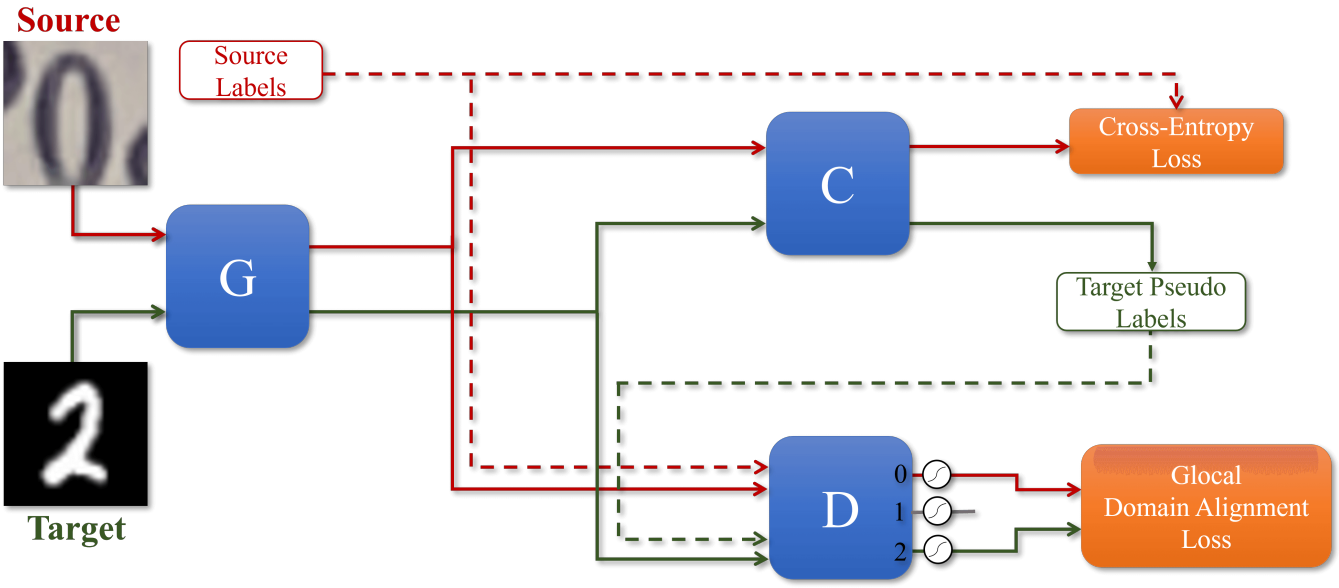

Figure 2: Model diagram of our approach. The flow of the source is denoted in Red and target using Green. The dotted lines indicate the flow of the labels. The generator G generates features that are classified by the classifier C and Glocally aligned by the discriminator D. The classifier C is trained using the cross-entropy loss on the source dataset and generates pseudo-labels for target samples. The discriminator D is a multi-headed binary classifier and trained using the Glocal domain-alignment loss function. The discriminator head D^i to train is selected using the input label and trained to classify between source and target's features belonging to i^{th} cateogry. The generator G uses both Glocal domain-alignment loss and cross-entropy loss for training. It is trained to fool the discriminator and minimize the source cross-entropy loss on the classification task. Best viewed in Color.

instead of D_t. Using these simple adjustments, any global alignment technique can be used at a local level.

The glocal loss functions are in Table 2. The discriminator head is selected based on the source label and target pseudo label. The generator is trained with the cross-entropy loss along with the glocal generator loss. The weighing coefficients for the generator are same as the traditional alignment losses.

5 EXPERIMENTS AND RESULTS

5.1 Datasets

We test our approach on the following classification tasks:

5.1.1 Digits and Traffic Signs. For digits experiments, we use 5 datasets - SVHN, MNIST, USPS, MNIST-M and Synthetic digits (SyDigits). MNIST is a handwritten digits dataset consisting of 28×28 grayscale images [8]. SVHN contains 32×32 real-colored images extracted from house numbers of Google Street View Images [14]. USPS is a 16×16 grayscale digit dataset obtained by scanning the digits from envelopes of the U.S. Postal Service. MNIST-M is a variation on MNIST created by reinforcing MNIST digits with the patches randomly extracted from color pictures of Berkeley Segmentation Data Set (BSDS500). Synthetic Digits (SynDigits) is a sythetic dataset of colored images with digits written in English font. All the digits datasets have 10 classes and present various visual variations in the images. Synthetic Signs (SynSigns) and GTSRB are traffic signs datasets containing 43 classes. The difference is that SySigns contains Wikipedia images whereas GTSRB has real-world traffic sign images [7].

We test our approach on the standard tasks: MNIST ↔ USPS, MNIST → MNIST-M, SVHN → MNIST, SynDigits → SVHN and SynSigns → GTSRB. Digits and Traffic sign images are scaled to 32×32 using bilinear interpolation and normalized to be in the range $[-1, 1]$. The G(Small) network is trained for these tasks (see below).

5.1.2 Office-31. Office-31 [16] is a real-world object classification dataset that contains 31 classes in three different domains – Amazon, DSLR and Webcam. Amazon domain contains images captured with a clean background; DSLR domain contains low-noise high resolution images and webcam domain has low resolution images with significant noise and color as white balance artifacts. The experiments are conducted using ResNet-50 pretrained on ImageNet. Standard random-crop and horizontal-flips are applied for training and center-crop for testing.

5.1.3 Office-Home. Office-Home [23] is another challenging real-world object classification dataset with 4 domains - Art (Ar), Clipart (Cl), Product (Pr) and Real-world (Rw) images. It consists of 15,500 images from 65 different categories that are found typically in Office and Home settings. ResNet-50 pretrained on ImageNet is used for

Method	MNIST→USPS	USPS→MNIST	MNIST→MNIST-M	SVHN→MNIST	SyDigits→SVHN	SySigns→GTSRB
Source	81.4	54.0	59.3	64.8	86.2	89.9
DANN[4]	93.5	96.2	83.2	75.3	91.7	92.3
$\mathcal{G}$-DANN	96.7$_\uparrow$	97.3$_\uparrow$	**85.3**$_\uparrow$	88.2$_\uparrow$	92.8$_\uparrow$	96.4$_\uparrow$
MDC[20]	96.3	97.9	–	72.8	91.9	93.7
$\mathcal{G}$-MDC	97.3$_\uparrow$	98.4$_\uparrow$	–	**89.7**$_\uparrow$	92.8$_\uparrow$	95.9$_\uparrow$
GAN_1[5]	96.8	98.1	81.7	65.9	92.3	95.2
$\mathcal{G}$-GAN_1	97.1$_\uparrow$	98.1	82.0$_\uparrow$	88.9$_\uparrow$	93.2$_\uparrow$	95.8$_\uparrow$
GAN_2[5]	97.1	98.3	81.9	73.1	92.5	93.6
$\mathcal{G}$-GAN_2	**97.2**$_\uparrow$	**98.7**$_\uparrow$	83.0$_\uparrow$	89.1$_\uparrow$	**93.4**$_\uparrow$	**97.1**$_\uparrow$

Table 3: Target classification accuracies on Digits and Traffic-Signs. The Glocal model is denoted as $\mathcal{G}$-(model). ("–" did not converge.)

Method	Amazon→Webcam	DSLR→Webcam	Webcam→DSLR	Amazon→DSLR	DSLR→Amazon	Webcam→Amazon	Mean
Source	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DANN[4]	82.5	97.6	99.6	81.5	67.9	71.7	83.5
$\mathcal{G}$-DANN	**92.6**$_\uparrow$	97.5$_\downarrow$	99.7$_\uparrow$	**89.8**$_\uparrow$	69.8$_\uparrow$	72.6$_\uparrow$	87.0$_\uparrow$
MDC[20]	87.0	97.4	**99.8**	83.3	70.0	72.7	85.0
$\mathcal{G}$-MDC	90.4$_\uparrow$	**98.5**$_\uparrow$	**99.8**	88.6$_\uparrow$	**73.7**$_\uparrow$	72.8$_\uparrow$	**87.3**$_\uparrow$
GAN_1[5]	85.5	97.1	**99.8**	84.3	68.4	72.4	84.6
$\mathcal{G}$-GAN_1	90.1$_\uparrow$	97.9$_\uparrow$	**99.8**	88.6$_\uparrow$	69.1$_\uparrow$	72.9$_\uparrow$	86.4$_\uparrow$
GAN_2[5]	85.5	97.2	**99.8**	83.1	69.7	72.8	84.7
$\mathcal{G}$-GAN_2	92.0$_\uparrow$	98.2$_\uparrow$	**99.8**	88.2$_\uparrow$	72.6$_\uparrow$	**73.2**$_\uparrow$	**87.3**$_\uparrow$

Table 4: Target classification accuracies for Office-31 using ResNet-50. The Glocal model is denoted as $\mathcal{G}$-(model).

these experiments. Standard random-crop and horizontal-flips are applied for training and center-crop for testing.

5.2 Training Setup

To ensure fair comparison, we train all the alignment loss approaches (including the baselines) using the following architecture:

G (Small): `K(32)`→`K(32)`→`P(2,2)`→`K(64)`→ `K(64)`→ `K(128)`→`K(128)`→`P(2,2)`→`FC(128)`

G (Office-31): `ResNet50`→`FC(512)`

G (Office-Home): `ResNet50`→`FC(512)`

$D(Global)$: `FC(500)`→`FC(500)`→`FC(1)`

$D(Glocal)$: `FC(500)`→`FC(500)`→`FC(`K`)`

C: `FC(`K`)`

`K(n)` represents `n` kernels of size 3 with padding 1, `P(2,2)` is a max pool with kernel size 2 and stride 2. `FC(n)` is a fully connected layer with `n` neurons. Classifier C and Discriminator D use the output of the generator G as the input. Classifier C is a one-layer network with K classes. Discriminator uses 1 neuron for global alignment and K neurons for local alignment. We use ReLU activation in the Generator and Softmax activation for the classifier. The discriminator uses Leaky ReLU with $\alpha = 0.2$ in the hidden layers and Sigmoid activation for the output.

The G(Small) and G(Office-31 & Office-Home) are trained with a batch size of 128 and 36 respectively. We use the Adam optimizer with 10^{-4} learning rate and βs = (0.5, 0.999). The learning rate for pretrained layers of ResNet-50 is set to 10^{-5} to ensure smooth fine-tuning. We set $\lambda_{g_1} = \lambda_{g_2} = \lambda_m = 0.01$, $\lambda_d = 1$ and $\tau = 0.9$ for all our experiments based on the existing literature. However, tuning τ should lead to further performance gains.

5.3 Target Classification Accuracy

Our experiment results are shown in Table 3, 4 and 5. In all cases, the Glocal alignment outperforms the category-agnostic global alignment. For the small-resolution images' experiments, the performance gain is the highest in case of SVHN→MNIST, which is the hardest adaptation task among them. For the diverse case of SynSigns→GTSRB with 43 classes, we did not experience any overfitting showcasing the strength of the multi-binary discriminator. In case of Office-31 and Office-Home experiments, Glocal alignment improves over global alignment with an average increase of 2.5%. The Glocal performance is comparable and in some cases is better than more complex approaches like ADDA [21], CDAN [11], TADA [24] and ALDA [3].

Source	Ar			Cl			Pr			Rw			Mean
Target	Cl	Pr	Rw	Ar	Pr	Rw	Ar	Cl	Rw	Ar	Cl	Pr	
Source	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN[4]	48.3	63.5	73.2	56.3	65.4	64.4	53.7	50.0	72.9	68.5	55.6	80.5	62.7
$\mathcal{G}$-DANN	**55.6**$_\uparrow$	68.0$_\uparrow$	75.7$_\uparrow$	**61.6**$_\uparrow$	68.6$_\uparrow$	**72.0**$_\uparrow$	**58.8**$_\uparrow$	**55.5**$_\uparrow$	78.4$_\uparrow$	**70.8**$_\uparrow$	**58.8**$_\uparrow$	**82.6**$_\uparrow$	**67.2**$_\uparrow$
MDC[20]	48.3	67.8	74.7	56.7	65.4	65.6	56.6	51.6	74.0	68.8	58.6	81.1	64.1
$\mathcal{G}$-MDC	55.3$_\uparrow$	**68.5**$_\uparrow$	75.5$_\uparrow$	57.9$_\uparrow$	**69.0**$_\uparrow$	70.5$_\uparrow$	56.2$_\downarrow$	54.0$_\uparrow$	78.2$_\uparrow$	69.9$_\uparrow$	**58.8**$_\uparrow$	82.1$_\uparrow$	66.3$_\uparrow$
GAN$_1$[5]	48.5	67.0	74.8	57.1	65.9	66.7	54.8	53.8	75.4	69.6	57.9	79.8	64.3
$\mathcal{G}$-GAN$_1$	53.5$_\uparrow$	66.5$_\downarrow$	75.3$_\uparrow$	57.4$_\uparrow$	67.5$_\uparrow$	68.9$_\uparrow$	55.0$_\uparrow$	54.5$_\uparrow$	75.8$_\uparrow$	69.8$_\uparrow$	57.5$_\downarrow$	80.8$_\uparrow$	65.2$_\uparrow$
GAN$_2$[5]	48.4	67.4	74.7	56.5	66.0	66.7	55.6	52.9	74.5	68.5	58.1	80.4	64.1
$\mathcal{G}$-GAN$_2$	55.3$_\uparrow$	68.3$_\uparrow$	**75.9**$_\uparrow$	58.3$_\uparrow$	68.2$_\uparrow$	71.4$_\uparrow$	57.4$_\uparrow$	54.7$_\uparrow$	**78.6**$_\uparrow$	69.9$_\uparrow$	58.4$_\uparrow$	81.7$_\uparrow$	66.5$_\uparrow$

Table 5: Results of Glocal domain alignment on Office-Home. The Glocal model is denoted as $\mathcal{G}$-(model).

5.4 A-distance

$\mathcal{A}$-distance is a metric to measure the domain gap, defined as $2\times(1-\varepsilon)$ where ε is the generalization error of a classifier trained to distinguish the features of the domains [1]. We perform 5-fold cross-validation using a linear SVM for GAN$_2$ on SVHN → MNIST transfer task. As depicted in Fig. 3, Global and Glocal alignments achieve similar low $\mathcal{A}$-distance (Fig. 3 (left)), which signifies that domains are well-aligned. However, when compared using the mean $\mathcal{A}$-distance for each category (Fig. 3 (right)), we observe a significant domain gap.

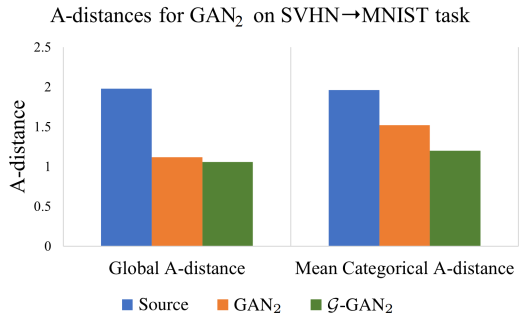

Figure 3: $\mathcal{A}$-distances for $\mathcal{G}$-GAN$_2$ on SVHN→ MNIST task.

5.5 Pseudo Label Accuracy

In almost all cases, the pseudo label accuracy of the mini-batch was similar to the mini-batch accuracy. Even with moderate pseudo label accuracy, the Glocal method achieves excellent performance. Fig. 4 shows training progress comparing pseudo-label accuracy with target accuracy for $\mathcal{G}$-GAN$_2$ on SVHN→MNIST task.

5.6 Domain Alignment Feature Visualization

We use t-SNE [13] plots to visualize feature alignment of the Glocal model for the SVHN→MNIST task in Fig. 5. Global alignment mixes the two domains well but also misaligns the individual categories,

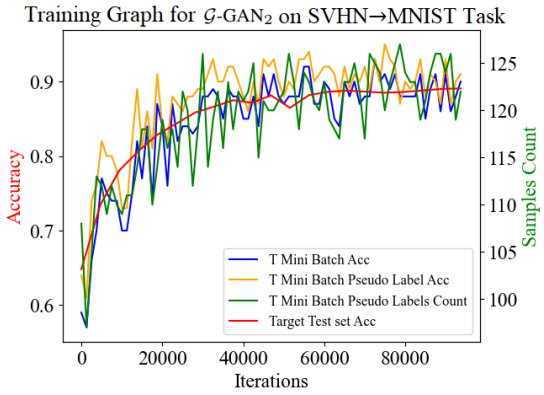

Figure 4: Training graphs comparing pseudo-label and target accuracies for $\mathcal{G}$-GAN$_2$ on SVHN→ MNIST task.

whereas our approach provides better alignment for individual categories along with the global alignment.

6 CONCLUSIONS

We presented the Glocal domain alignment technique with a salient modification to global alignment loss functions. Glocal alignment uses the most confident target pseudo labels and aligns individual categories which in turn improves global alignment. Through extensive experiments on various small and large datasets, we showcase the strength of the Glocal alignment. In all the cases, Glocal alignment results in superior performance compared to Global adversarial alignment.

ACKNOWLEDGMENTS

The work was supported in part by a grant from ONR. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR.

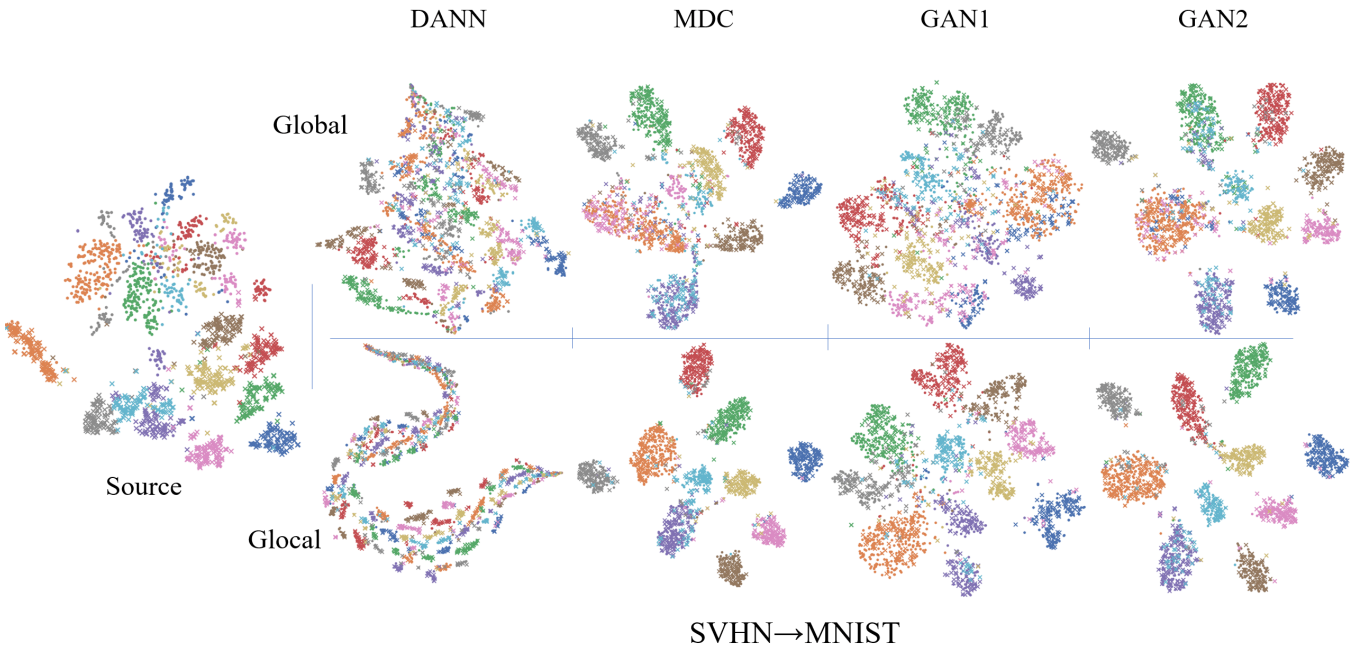

Figure 5: t-SNE plots for SVHN→MNIST task. Each color represents a class. Source is represented by • and target by +.

REFERENCES

[1] 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, 1-2 (2010), 151–175.

[2] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. 2019. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 627–636.

[3] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. 2020. Adversarial-learned loss for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3521–3528.

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[5] 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*. 2672–2680.

[6] 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 *International conference on machine learning*. PMLR, 1989–1998.

[7] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. 2013. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[9] Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. 2.

[10] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. 97–105.

[11] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. In *Advances in neural information processing systems*. 1640–1650.

[12] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*. PMLR, 2208–2217.

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

[14] 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 *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

[15] Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR* abs/1706.05098 (2017). arXiv:1706.05098 http://arxiv.org/abs/1706.05098

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

[17] 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*.

[18] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. 2018. A DIRT-T Approach to Unsupervised Domain Adaptation. In *International Conference on Learning Representations*.

[19] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A Survey on Deep Transfer Learning. In *International conference on artificial neural networks*. Springer, 270–279.

[20] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. 2015. Simultaneous Deep Transfer Across Domains and Tasks. In *Proceedings of the IEEE International Conference on Computer Vision*. 4068–4076.

[21] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.

[22] Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations. *IEEE Signal Processing Magazine* 34, 6 (2017), 117–129.

[23] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5018–5027.

[24] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. 2019. Transferable attention for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5345–5352.

[25] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. 2018. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*. 5423–5432.