DOMAIN ADAPTATION THROUGH LABEL PROPAGATION: LEARNING CLUSTERED AND ALIGNED FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

The difficulty of obtaining sufficient labeled data for supervised learning has motivated domain adaptation, in which a classifier is trained in one domain, *source domain*, but operates in another, *target domain*. Reducing domain discrepancy has improved the performance, but it is hampered by the embedded features that do not form clearly separable and aligned clusters. We address this issue by propagating labels using a manifold structure, and by enforcing cycle consistency to align the clusters of features in each domain more closely. Specifically, we prove that cycle consistency leads the embedded features distant from all but one clusters if the source domain is ideally clustered. We additionally utilize more information from approximated local manifold and pursue local manifold consistency for more improvement. Results for various domain adaptation scenarios show tighter clustering and an improvement in classification accuracy.

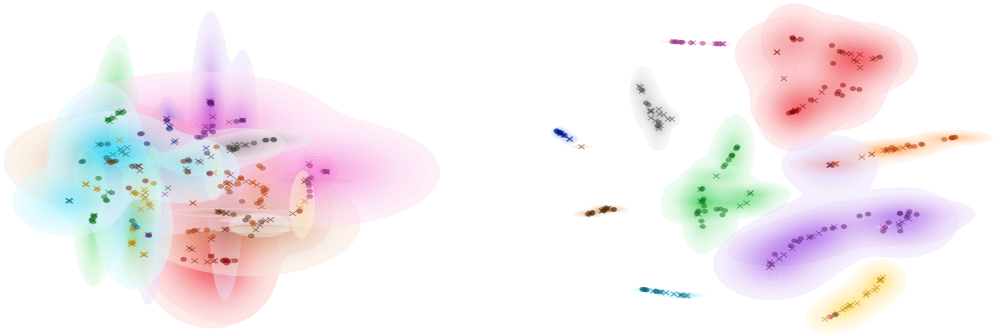
1 INTRODUCTION

Classifiers trained through supervised learning have many applications (Bahdanau et al., 2015; Redmon et al., 2016), but it requires a great deal of labeled data, which may be impractical or too costly to collect. Domain adaptation circumvents this problem by exploiting the labeled data available in a closely related domain. We call the domain where the classifier will be used at, the target domain, and assume that it only contains unlabeled data $\{x^t\}$; and we call the closely related domain the source domain and assume that it contains a significant amount of labeled data $\{x^s, y^s\}$.

Domain adaptation requires the source domain data to share discriminative features with the target data (Pan et al., 2010). In spite of the common features, a classifier trained using only the source data is unlikely to give satisfactory results in the target domain because of the difference between two domains' data distributions, called *domain shift* (Pan et al., 2010). This may be addressed by *fine-tuning* on the target domain with a small set of labeled target data, but it tends to overfit to the small labeled dataset (Csurka, 2017).

Another approach is to find discriminative features which are invariant between two domains by reducing the distance between the feature distributions. For example, domain-adversarial neural network (DANN) (Ganin et al., 2016) achieved remarkable result using generative adversarial networks (GANs) (Goodfellow et al., 2014). However, this approach still has room to be improved. Because the classifier is trained using labels from the source domain, the source features become clustered, and they determine the decision boundary. It would be better if the embedded features from the target domain formed similar clusters to the source features in class-level so that the decision boundary does not cross the target features. Methods which only reduce the distance between two marginal distributions bring the features into general alignment, but clusters do not match satisfactorily, as shown in Fig. 1(a). As a consequence, the decision boundary is likely to cross the target features, impairing accuracy.

In this work, we propose a novel domain adaptation method to align the manifolds of the source and the target features in class-level, as shown in Fig. 1(b). We first employ label propagation to evaluate the relation between manifolds. Then, to align them, we reinforce the cycle consistency that is the correspondence between the original labels in the source domain and the labels that are propagated from the source to the target and back to the source domain. The cycle consistency draws features



(a) DANN model (b) Our model

Figure 1: Features from the SVHN $\rightarrow$ MNIST scenario visualized using t-SNE. Circle and x markers represent the source and target domain features, respectively. In (a), features from DANN model are aligned but the fit is far from perfect, and the boundaries between classes are not clear. In contrast, our model in (b) produces clearly aligned and clustered features.

from both domains that are near to each other to converge, and those that are far apart to diverge. The proposed method exploits manifold information using label propagation which had not been taken into account in other cycle consistency based methods. As a result, our approach outperforms other baselines on various scenarios as demonstrated in Sec. 4. Moreover, the role of cycle consistency is theoretically explained in Sec. 3.2 that it leads to aligned manifolds in class-level. To acquire more manifold information within the limited number of mini-batch samples, we utilize local manifold approximation and pursue local manifold consistency. In summary, our contributions are as follows:

- We propose a novel domain adaptation method which exploits global and local manifold information to align class-level distributions of the source and the target.
- We analyze and demonstrate the benefit of the proposed method over the most similar baseline, Associative domain adaptation (AssocDA) (Haeusser et al., 2017).
- We present the theoretical background on why the proposed cycle consistency leads to class-level manifold alignment, bringing better result in domain adaptation.
- We conduct extensive experiments on various scenarios and achieve the state-of-the-art performance.

2 Related Work

Unsupervised Domain Adaptation It has been shown (Ben-David et al., 2010) that the classification error in the target domain is bounded by that in the source domain, the discrepancy between the domains and the difference in labeling functions. Based on this analysis, a number of works have endeavored to train domain-confusing features to minimize the discrepancy between the domains (Ganin et al., 2016; Long et al., 2013; 2015; Tzeng et al., 2014; 2017). Maximum mean discrepancy can be used (Long et al., 2015; Tzeng et al., 2014) as a measure of domain discrepancy. In an approach inspired by GANs, a domain confusion can be converted (Ganin et al., 2016; Tzeng et al., 2017) into a minmax optimization.

While minimization of domain discrepancy can be effective in reducing the upper bound on the error, it does not guarantee that the feature representation in the target domain is sufficiently discriminative. To address this issue, several techniques had been proposed. Explicit separation of the shared representation from the individual characteristics of each domain may enhance the accuracy of the model (Bousmalis et al., 2016). This approach has been implemented as a network with private and shared encoders and a shared decoder. The centroid and prototype of each category can be used for class-level alignment (Pinheiro, 2018; Xie et al., 2018). An alternative to such feature-space adaptation techniques is the direct conversion of target data to source data (Bousmalis et al., 2017; Hoffman et al., 2018; Yoo et al., 2017). Those proposed methods intend to transfer the style of images to another domain while preserving the content. This performs well on datasets containing

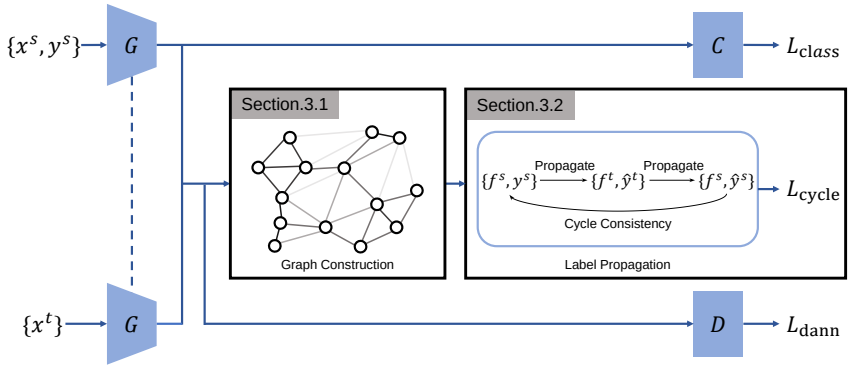


Figure 2: Overview of our method. The feature generator G projects the input data into the feature space. The dashed line means weight sharing. The embedded source features f^s and the target features f^t are organized into a graph and then used together to evaluate cycle consistency through label propagation. The embedding classifier C learns from the source ground-truth labels. The discriminator D determines whether features originated in the source or the target domain.

images that are similar at the pixel-level; they are problematic when the mapping between high-level features and images is complicated (Tzeng et al., 2017).

Metric Learning Metric learning is learning an appropriate metric distance to measure the similarity or dissimilarity between data (Bellet et al., 2013). Reducing the distances between similar data and increasing the distances between distinct data has shown (Schroff et al., 2015) to improve the accuracy of a classifier.

Metric learning is particularly beneficial when very little labeled data is available, which is the situation for domain adaptation. Sener et al. (2016) combined metric learning and unsupervised domain adaptation with the enforcement of cycle consistency. In particular, the inner products of source features and target features with the same label are maximized, and minimized between features with different labels. AssocDA (Haeusser et al., 2017) enforces the feature alignment between the source and target by forcing the two step round trip probability to be uniform in the same class and to vanish between different classes.

Graph-based learning is closely related to metric learning, in that it achieves clustering using distance information. Label consistency (Zhou et al., 2004) is usually assumed, meaning that adjacent data tend to have the same labels (Wang et al., 2009). Label propagation (Zhou et al., 2004) has improved the performance of semi-supervised learning by enforcing label consistency by propagating labels from labeled to unlabeled data. To overcome need for fixed graphs to be provided in advance, the distances between each node can be adaptively learned (Liu et al., 2019; Oshiba et al., 2019), as in metric learning, and this increases accuracy in both semi-supervised and few-shot learning.

3 METHOD

Our algorithm, shown in Fig. 2, uses label propagation and cycle consistency to learn features from the source and the target domains which are both 1) indistinguishable each other and 2) close when placed within the same class, but distant when placed in different classes. The details are as follows.

3.1 FEATURE EMBEDDING AND GRAPH CONSTRUCTION

Manifold learning (Nie et al., 2010) extracts intrinsic structures from both unlabeled and labeled data. We obtain these structures by constructing a graph whose vertexes are the embedded features and whose edges are the relations between data. We first embed the input data in the feature space, using the feature generator composed of convolutional layers following previous work (Liu et al., 2019; Oshiba et al., 2019). Subsequently, a fully connected graph is constructed according to the distances between the features. The edge weights W_{ij} between the input data x_i, x_j are determined from the feature vectors using Gaussian similarity, $W_{ij} = \exp(-\frac{\|f_i - f_j\|^2}{2\sigma^2})$, where f_i, f_j are the embedded feature vectors of x_i, x_j, and σ is a scale parameter. It is known (Liu et al., 2019) that

graph-based methods are sensitive to the scale parameter σ. A large σ results in an uniformly connected graph that disregards the latent structure of the data, while a small σ produces a sparse graph which fails to express all the relationship between the data. To adapt σ according to the embedded features, we take σ as a trainable variable to be learned during training.

3.2 Label Propagation and Cycle Consistency

Label propagation (Zhou et al., 2004) is a method of manifold regularization, which in turn produces a classifier that is robust against small perturbations. Label propagation can be seen as a repeated random walk through the graph of features using an affinity matrix to assign the labels of target data (Xiaojin & Zoubin, 2002).

A label matrix $y_n \in \mathbb{R}^{(\mathrm{N}_s+\mathrm{N}_t)\times \mathrm{C}}$ refers to the labels assigned to data in both domains at the n-th step random walk. The dimension of y_n is determined by N_s, N_t, and C which are the numbers of source and target data points and the number of classes, respectively. The first N_s rows of y_n contain the labels of the source data, and the remaining N_t rows contain the labels of the target data. The initial label vector y_0 contains y^s for the source data, which is one-hot coded ground-truth labels and zero vectors for the target data.

The one step of the random walk transforms the label vector as follows:

$$y_{n+1} = T y_n. \tag{1}$$

where, $T = \begin{pmatrix} I & 0 \\ T_{ts} & T_{tt} \end{pmatrix} = \mathrm{normalize}(W)$ and $W = \begin{pmatrix} I & 0 \\ W_{ts} & W_{tt} \end{pmatrix}$. W_{ts} is a similarity matrix between the target and source data, and W_{tt} is a similarity matrix which represents the interrelations in the target data. These are described in the Sec. 3.1. The normalization operation $\mathrm{normalize}(\cdot)$ transforms the sum of each row to 1. The identity matrix in the normalized transition matrix T signifies that the labels of source data do not change because its labels are already known. In graph theory, these source data points would be called absorbing nodes.

In label propagation, the labels of the target domain is assigned to the propagated labels $\hat{y}^t$ by infinite transition, formulated as $\hat{y}^t = \lim_{n\to\infty} \sum_{i=1}^{n} T_{tt}^{i-1} T_{ts} y^s$, which converges as follows (Xiaojin & Zoubin, 2002):

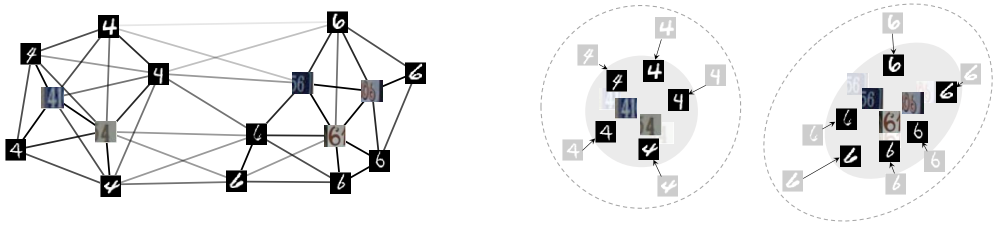
$$\hat{y}^t = (I - T_{tt})^{-1} T_{ts} y^s. \tag{2}$$

In our method, $\hat{y}^t$ is used to obtain the propagated labels of the source data in the same way as $\hat{y}^s = (I - T_{ss})^{-1} T_{st} \hat{y}^t$ where T_{ss} and T_{st} are defined analogous to T_{tt} and T_{ts}, so that we can learn the features of which clusters match each other. We then refer to the property that $\hat{y}^s$ should be the same as the original label y^s as cycle consistency. Pursuing cycle consistency forces not perfectly aligned features to move toward the nearest cluster, as shown in Fig. 3. The following theorem shows that enforcing cycle consistency on ideally clustered source data will segregate different classes of the source and the target data and gather the same classes.

Theorem 1. *Let $\{e_i | 1 \le i \le \mathrm{C}\}$ be the standard bases of C-dimensional Euclidean space. For the sake of simplicity, source data $x_1, x_2, \cdots, x_{\mathrm{N}_s}$ are assumed to be arranged so that the first n_1 data belong to class 1, the n_2 data to class 2, and so forth. Assume that 1) the source data is ideally clustered, in the sense that T_{ss} has positive values if the row and the column are the same class and zero otherwise, i.e., $T_{ss} = diag(T_1, T_2, \cdots, T_{\mathrm{C}})$, the block diagonal where T_i is a $n_i \times n_i$ positive matrix for $i = 1, 2, \cdots, \mathrm{C}$ and 2) $\hat{y}^s = y^s$. Then for all $1 \le j \le \mathrm{C}$, there exists a nonnegative vector $v_j \in \mathbb{R}^{\mathrm{N}_s}$ such that 1) the part where source data belongs to j^{th} class (from $[n_1 + n_2 + \cdots + n_{j-1} + 1]^{\mathrm{th}}$ element to $[n_1 + n_2 + \cdots + n_j]^{\mathrm{th}}$ element) are positive and the other elements are all zero and 2) $v_j^\top T_{st} \hat{y}^t e_i = 0$ for all $1 \le i \le \mathrm{C}, i \ne j$.*

Proof. The illustration and the proof is given in Appx. A. □

In Thm. 1, $\hat{y}^t e_i$ refers to the assigned probability as i^{th} class to the target data. The conclusion implies that if a target data is enough to be predicted as i^{th} class through label propagation, *i.e.*, i^{th} elements of the row in $\hat{y}^t$ corresponding to the target data is nonzero, then the elements of T_{st} which represent the transitions from source data of all but i^{th} class to the target data should



(a) Graph construction (b) Effect of cycle consistency

Figure 3: Graphical interpretation of the effect of cycle loss in the SVHN→MNIST scenario. (a) The model constructs a graph in the feature space, and the darkness of each line is proportional to similarity of the features. (b) Features with high similarity, expressed as both direct and indirect connections, cluster together by class to enforce cycle consistency.

vanish, *i.e.*, the target data is segregated from the source data in different classes. As described in Sec. 3.4, we employed DANN to prevent the target data distribution to be distinct from the source data distribution. If a column of T_{st} is a zero vector, the feature of the corresponding target data for the column is considerably distant from all source data features. However, minimizing the DANN loss makes target features lie around source features, and thus each column of T_{st} is not likely to be a zero vector. Combining this conjecture with Thm. 1, each row of $\hat{y}^t$ has only one nonzero value, *i.e.*, every target data belongs to only one cluster. We thus argue that by pursuing this property, generator can learn more discriminative shared features, and classification performance may improve. Cycle consistency is enforced by minimizing the l_1 loss L_{cycle} between $\hat{y}^s$ and y^s:

$$L_{\mathrm{cycle}} = \left\| \hat{y}^s - y^s \right\|_1 . \tag{3}$$

Comparison with AssocDA The proposed method has some resemblance with AssocDA in that they both consider the similarities and transitions between data. However, we argue that AssocDA is a special case of our method. First, our method exploits manifold over each domain by taking relations within the same domain into account through label propagation, whereas AssocDA only considers relations across the domains. Specifically, in Eq. 1, our method utilizes both T_{ts} and T_{tt}, but AssocDA ignores T_{tt} which often has useful information about the target data manifold. Second, AssocDA forces the two-step transition to be uniform within the same class. This strict condition may drive the source features of each class to collapse to one mode and can cause overfitting. On the contrary, our method only constrains source data to preserve its original labels after the label propagation. Thus, it does not require all source data be close to each other within the same class; it allows moderate intra-class variance. The experiment in Sec. 4.1 and Fig. 4 support these arguments and visualize the effect of the differences.

3.3 Local manifold consistency

As shown in Thm. 1, the introduced cycle consistency utilizes graph based global manifold information and enforces the source and target features to be aligned in class-level. However, in practice, the limited size of mini-batch may restrict the available information of graph. The knowledge from the local manifold of each sample, in this case, can complement the global manifold information. In this regard, we additionally pursue local manifold consistency that the output should not be sensitive to small perturbations in the local manifold, as suggested elsewhere (Simard et al., 1992; Kumar et al., 2017; Qi et al., 2018). Concretely, localized GAN (LGAN) (Qi et al., 2018) is employed to approximate the local manifold of each data and sample a marginally perturbed image along the local manifold from the given data. LGAN allows it as LGAN focuses on learning and linking patches of local manifolds in its training procedure. The difference between the predicted label of the perturbed image and that of the original image is minimized to impose local manifold consistency of the classifier as follows:

$$\begin{aligned} L_{\mathrm{local}} = \mu \frac{1}{\mathrm{N}_s} \sum_{i=1}^{\mathrm{N}_s} \mathbb{E}_{\mathrm{z}\sim P_z} H\big(C(G(x_i^s)), C(G(G_L(x_i^s, z)))\big) \\ + \eta \frac{1}{\mathrm{N}_t} \sum_{j=1}^{\mathrm{N}_t} \mathbb{E}_{\mathrm{z}\sim P_z} H\big(C(G(x_j^t)), C(G(G_L(x_j^t, z)))\big), \end{aligned} \tag{4}$$

where, C, G and G_L are the embedding classifier, the feature generator and the LGAN generator, respectively. LGAN generator, $G_L(x, z)$, takes an image x and noise z to generate locally perturbed image along the approximated local manifold. $H(\cdot, \cdot)$ denotes cross entropy. μ and η are coefficients for the source and the target local manifold consistency loss, respectively.

3.4 Training Process

Our method learns a clustered feature representation that is indistinguishable across the source and target domains through the training process as follows:

$$\max_{D} \quad L_{\text{dann}} \tag{5}$$

$$\min_{G,C} \; L_{\text{class}} + \lambda\alpha L_{\text{dann}} + \lambda\beta L_{\text{cycle}} + L_{\text{local}}, \tag{6}$$

where, D is the discriminator. α and β are coefficients for the last two terms and λ is a scheduling parameter described in Appx B.1. L_{class} is a widely used cross-entropy loss for labeled source data and L_{dann} is a GAN loss (Ganin et al., 2016; Goodfellow et al., 2014):

$$L_{\text{class}} = \frac{1}{\mathrm{N}_s} \sum_{i=1}^{\mathrm{N}_s} -\log p_i(y = y_i^s) \tag{7}$$

$$L_{\text{dann}} = \frac{1}{\mathrm{N}_s} \sum_{i=1}^{\mathrm{N}_s} \log D(G(x_i^s)) + \frac{1}{\mathrm{N}_t} \sum_{j=1}^{\mathrm{N}_t} \log\left(1 - D(G(x_j^t))\right), \tag{8}$$

where discriminator's output $D(\cdot)$ is the probability that the input originated from the source domain. From the metric learning perspective, L_{class} serves to separate the source features according to their ground-truth labels, which supports the assumption in Thm. 1, the ideally clustered source features. Subsequently, L_{dann} takes a role in moving the target features toward the source features, but it is insufficient to produce perfectly aligned clusters. Our cycle loss L_{cycle} and local loss L_{local} facilitate clustering by enforcing cycle consistency and local manifold consistency.

4 Experiments

4.1 Toy Example

We present a toy example to empirically demonstrate the effect of our proposed cycle loss using manifold information compared to the most similar method, AssocDA. We designed synthetic dataset in 2-dimensional feature space with two classes as illustrated in the leftmost of Fig. 4. The source data lie vertically and the target data are slightly tilted and translated. The second column shows the negative gradients of AssocDA loss and our cycle loss with respect to each data. Negative gradients can be interpreted as the movement of features at each iteration. The third and fourth are the updated features using gradient descent in the middle and at the end of feature updates[1].

As argued in Sec. 3.2, AssocDA does not consider the transition within the same domain and thus target data which are close to source data with different label (points inside red circles in the second column) are strongly attracted to them. On the other hand, the gradients of the cycle loss are much smaller than AssocDA. We speculate that it is because the attractions from source data in the same class are propagated through target data manifold. As a result, AssocDA leads some data to move in wrong direction, being misclassified, while cycle loss brought correctly aligned manifolds. In addition, AssocDA attracts all features too close at the end of updates, which may cause overfitting. Last but not least, our cycle loss aligned source and target clusters correctly without the aid of dann loss. We thus argue that our method is complementary to DANN rather than an extension.

4.2 Real dataset experiment

We show the performance of the proposed method on two real visual dataset. First dataset, which we call by *Digit & Object* dataset, includes digit dataset such as SVHN and Synthetic Digits (DIGITS),

[1]The animation of update progress is available at https://youtu.be/09PE5iXwvzY

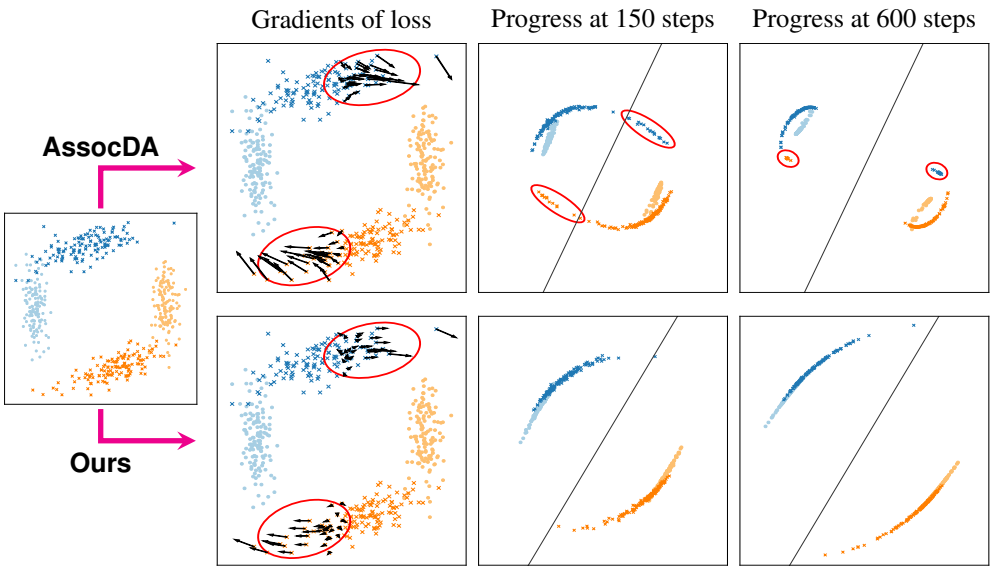
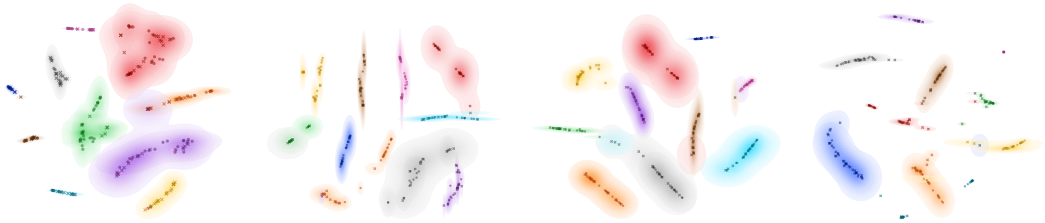


Figure 4: Visualization of toy experiment. (Best viewed in color.) Blue and orange colors represent labels. Circles with light color are source data and x markers with dark color are target data. The left-most one depicts the initial data distribution. For the right six sub-figures, the top row refers to AssocDA and the bottom row refers to ours. The second column illustrates the negative gradients of loss for target data that are close to the source with different labels. The third and fourth columns are the updated data after gradient descent in the middle and at the end of the training. The black lines indicate the decision boundaries of logistic regression models trained with source labels. Ours aligns manifolds better than AssocDA and results in an accurate classifier for the target.



(a) SVHN → MNIST (b) USPS → MNIST (c) MNIST → MNIST-M (d) MNIST → USPS

Figure 5: Visualization of learned features using t-SNE. Circles and x markers respectively indicate the source and target features. Colors correspond to labels. In all cases, the features from two domains form similar and tight clusters, which is the key objective of our method.

and object dataset such as STL and CIFAR. We used ImageCLEF-DA as second dataset for more challenging benchmark. We employed three networks as previous work (Shu et al., 2018; Xie et al., 2018; Long et al., 2018). A network with two convolutional layers and two fully connected layers for digit dataset and a network with nine convolutional layers and one fully connected layer for object dataset were implemented. Pretrained ResNet (He et al., 2016) was used for ImageCLEF-DA dataset. More details on training settings, adaptation scenarios and an experiment on non-visual dataset are provided in Appx. B.1, B.2 and E.

Tab. 1 compares the accuracy of our method on Digit & Object dataset with that of other approaches. For our method, we reported the results of three models, one with local loss (L), another with cycle loss (C) and the other with both losses (C+L). Our algorithm outperformed the others on most of the tasks. In the most experiments, the performance of the proposed method was better than the state-of-the-art. This suggests that enforcing alignment in addition to domain-invariant embedding reduces the error-rate. PixelDA (Bousmalis et al., 2017) showed superior performance on MNIST→MNIST-M, but it is attributable to the fact that PixelDA learns transferring the style of images at a pixel level which is similar (Pinheiro, 2018) to the way MNIST-M is generated from MNIST. T-SNE embeddings in Fig. 5 indicates that the learned features are well aligned and clustered.

Table 1: Accuracy (%) on Digit & Object dataset. Most results are excerpted from (Bousmalis et al., 2017; Tzeng et al., 2017). All experiments were run 5 times.

Source Target	MNIST MNIST-M	MNIST USPS	USPS MNIST	SVHN MNIST	DIGITS SVHN	CIFAR STL	STL CIFAR
Source Only	63.6	75.2	57.1	60.1	86.9	76.3	63.6
DANN (Ganin et al., 2016)	76.7	77.1	73.0	73.9	90.3	-	-
DRCN (Ghifary et al., 2016)	-	91.8	73.7	82.0	-	66.4	58.7
CoGAN (Liu & Tuzel, 2016)	62.0	91.2	89.1	-	-	-	-
DSN (Bousmalis et al., 2016)	83.2	-	-	82.7	91.2	-	-
JAN (Long et al., 2017)	76.9	81.1	-	71.1	88.0	-	-
ADDA (Tzeng et al., 2017)	-	-	-	76.0	-	-	-
AssocDA (Haeusser et al., 2017)	89.5	-	-	97.6	91.9	-	-
PixelDA (Bousmalis et al., 2017)	**98.2**	95.9	-	-	-	-	-
ATT (Saito et al., 2017)	94.2	-	-	86.2	92.9	-	-
LEL (Luo et al., 2017)	-	-	-	81.0	-	-	-
CyCADA (Hoffman et al., 2018)	-	95.6	96.5	90.4	-	-	-
SimNet (Pinheiro, 2018)	90.5	96.4	95.6	-	-	-	-
MSTN (Xie et al., 2018)	-	-	-	91.7	-	-	-
MCD (Saito et al., 2018)	-	-	-	96.2	-	-	-
CDAN+E (Long et al., 2018)	-	95.6	98.0	89.2	-	-	-
VADA† (Shu et al., 2018)	91.1	91.3	91.4	93.1	89.8	80.0	75.3
DIRT-T† (Shu et al., 2018)	93.7	90.5	93.3	n.c†	90.0	-	-
PFAN (Chen et al., 2019)	-	-	-	93.9	-	-	-
rRevGrad+CAT (Deng et al., 2019)	-	-	-	**98.8**	-	-	-
MCD+CAT (Deng et al., 2019)	-	-	-	97.1	-	-	-
Ours (L)	91.2±0.8	95.9±0.3	97.6±0.3	76.2±8.1	91.9±0.2	80.1±0.8	**75.8±0.4**
Ours (C)	96.5±0.1	**97.3±0.2**	98.6±0.1	98.2±0.2	92.1±0.2	80.5±0.3	69.9±0.3
Ours (C+L)	96.4±0.1	97.2±0.2	**99.2±0.1**	98.2±0.1	**93.4±0.1**	**81.4±0.5**	75.6±0.4

† Results on VADA and DIRT-T for all but CIFAR↔STL experiment are obtained by running publicly available code with a modification of network to be same with ours for a fair comparison. In SVHN → MNIST experiment, DIRT-T did not converge and collapsed.

Table 2: Accuracy (%) on ImageCLEF-DA for domain adaptation tasks

Source → Target	I → P	P → I	I → C	C → I	C → P	P → C	Avg
Source Only	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN (Long et al., 2015)	74.5±0.4	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
DANN (Ganin et al., 2016)	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
JAN (Long et al., 2017)	76.8±0.4	88.0±0.2	94.7±0.2	89.5±0.3	74.2±0.3	91.7±0.3	85.8
CDAN (Long et al., 2018)	76.7±0.3	90.6±0.3	97.0±0.4	90.5±0.4	74.5±0.3	93.5±0.4	87.1
CDAN+E (Long et al., 2018)	77.7±0.3	90.7±0.2	**97.7±0.3**	**91.3±0.3**	74.2±0.2	94.3±0.3	87.7
MADA (Pei et al., 2018)	75.0±0.3	87.9±0.2	96.0±0.3	88.8±0.3	75.2±0.2	92.2±0.3	85.8
LAD (Manders et al., 2019)	76.8±0.7	90.6±0.6	95.2±0.3	88.5±1.0	74.0±1.0	94.1±0.2	86.5
CAT (Deng et al., 2019)	76.7±0.2	89.0±0.7	94.5±0.4	89.8±0.3	74.0±0.2	93.7±1.0	86.3
JAN+CAT (Deng et al., 2019)	76.3±0.8	89.2±0.8	95.3±0.7	89.3±0.3	75.9±1.1	92.2±1.3	86.4
rRevGrad+CAT (Deng et al., 2019)	77.2±0.2	91.0±0.3	95.5±0.3	**91.3±0.3**	75.3±0.6	93.6±0.5	87.3
Ours (C)	**78.1±0.5**	**91.8±0.5**	96.4±0.5	90.6±1.1	**76.3±0.9**	**95.7±0.6**	**88.2**
Ours (C+L)	77.7±0.6	91.3±0.7	95.8±0.3	89.9±0.5	76.0±0.4	95.4±0.8	87.7

Tab. 2 reports the results on ImageCLEF-DA dataset experiments. The performance of our method was better than or comparable to those of other baselines. Especially, our method outperforms CAT Deng et al. (2019) which also aims to learn clustered and aligned features. Although the objectives are related, the approaches are quite different. Our method utilizes the manifolds of the source and the target domain through label propagation and cycle consistency, whereas CAT considers the distance between two samples for clustering and the distance between the first-order statistics of distributions for alignment. We argue that the better performance is attributed to utilizing manifold information beyond one to one relations of which benefits are explained in Sec. 4.1. Throughout ImageCLEF-DA experiments, the proposed method without the local loss achieved better accuracy compared to that with the local loss. Approximation of the local manifold on ImageCLEF-DA generated by LGAN was slightly worse than that on Digit & Object dataset; perturbed image was blurred and semantically invariant with the original image. Hence, we speculate that the performance of the proposed method may be improved with better local manifold approximation.

5 CONCLUSION

In this paper, we proposed a novel domain adaptation which stems from the objective to correctly align manifolds which might result in better performance. Our method achieved it, which was supported by intuition, theory and experiments. In addition, its superior performance was demonstrated on various benchmark dataset. Based on graph, our method depends on how to construct the graph. Pruning the graph or defining a similarity matrix considering underlying geometry may improve the performance. Our method also can be applied to semi supervised learning only with slight modification. We leave them as future work.

REFERENCES

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447, 2007.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 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)*, pp. 7, 2017.

Chaoqi Chen, Weiping Xie, Tingyang Xu, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

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

Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

John S Denker, WR Gardner, Hans Peter Graf, Donnie Henderson, Richard E Howard, W Hubbard, Lawrence D Jackel, Henry S Baird, and Isabelle Guyon. Neural network recognizer for handwritten zip code digits. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 323–331, 1989.

Georg Frobenius, Ferdinand Georg Frobenius, Ferdinand Georg Frobenius, Ferdinand Georg Frobenius, and Germany Mathematician. Über matrizen aus nicht negativen elementen. 1912.

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 (ECCV)*, 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 (NIPS)*, pp. 2672–2680, 2014.

Anne Greenbaum, Ren-cang Li, and Michael L Overton. First-order perturbation theory for eigenvalues and eigenvectors. *arXiv preprint arXiv:1903.00785*, 2019.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pp. 1989–1998, 2018.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: manifold invariance with improved inference. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5534–5544, 2017.

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.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 469–477, 2016.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagating network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2200–2207, 2013.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 97–105, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2208–2217. JMLR. org, 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variaional fair autoencoder. In *Internation Conference on Learning Representation (ICLR)*, 2016.

Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 165–177, 2017.

Jeroen Manders, Twan van Laarhoven, and Elena Marchiori. Adversarial alignment of class prediction uncertainties for domain adaptation. pp. 221–231, 01 2019. doi: 10.5220/0007519602210231.

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*, pp. 5, 2011.

Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932, 2010.

Kojin Oshiba, Nir Rosenfeld, and Amir Globerson. Label propagation networks, 2019. URL `https://openreview.net/forum?id=r1g7y2RqYX`.

Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

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

Oskar Perron. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907.

Pedro O Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8004–8013, 2018.

Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1517–1525. IEEE, 2018.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 2988–2997, 2017.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2110–2118, 2016.

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2018.

Patrice Simard, Bernard Victorri, Yann LeCun, and John Denker. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In *Advances in neural information processing systems*, pp. 895–903, 1992.

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)*, pp. 4, 2017.

Jingdong Wang, Fei Wang, Changshui Zhang, Helen C Shen, and Long Quan. Linear neighborhood propagation and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1600–1615, 2009.

Zhu Xiaojin and Ghahramani Zoubin. Learning from labeled and unlabeled data with label propagation. *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002.

Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 5419–5428, 2018.

Jaeyoon Yoo, Yongjun Hong, and Sungrho Yoon. Autonomous uav navigation with domain adaptation. *arXiv preprint arXiv:1712.03742*, 2017.

Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 321–328, 2004.

A PROOF OF THEOREM 1

Theorem 1. *Let* $\{e_i | 1 \leq i \leq \mathrm{C}\}$ *be the standard bases of* C*-dimensional Euclidean space. For the sake of simplicity, source data* $x_1, x_2, \cdots, x_{\mathrm{N}_s}$ *are assumed to be arranged so that the first* n_1 *data belong to class* 1*, the* n_2 *data to class* 2*, and so forth. Assume that 1) the source data is ideally clustered, in the sense that* T_{ss} *has positive values if the row and the column are the same class and zero otherwise, i.e.,* $T_{ss} = diag(T_1, T_2, \cdots, T_{\mathrm{C}})$*, the block diagonal where* T_i *is a* $n_i \times n_i$ *positive matrix for* $i = 1, 2, \cdots, \mathrm{C}$ *and 2)* $\hat{y}^s = y^s$*. Then for all* $1 \leq j \leq \mathrm{C}$*, there exists a nonnegative vector* $v_j \in \mathbb{R}^{\mathrm{N}_s}$ *such that 1) the part where source data belongs to* j^{th} *class (from* $[n_1 + n_2 + \cdots + n_{j-1} + 1]^{\mathrm{th}}$ *element to* $[n_1 + n_2 + \cdots + n_j]^{\mathrm{th}}$ *element) are positive and the other elements are all zero and 2)* $v_j^\top T_{st} \hat{y}^t e_i = 0$ *for all* $1 \leq i \leq \mathrm{C}, i \neq j$.

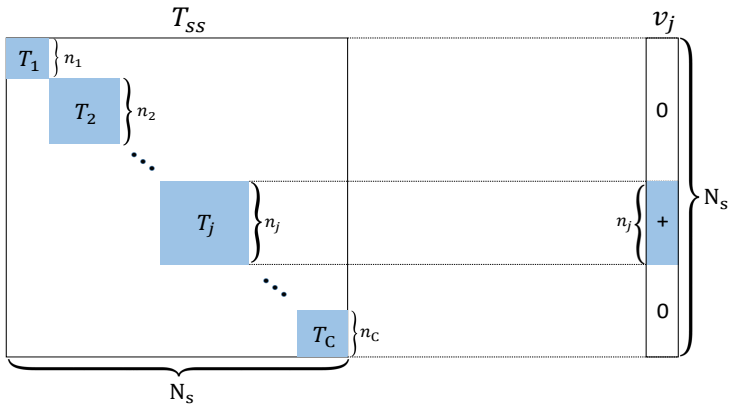


Figure 6: Illustration for Theorem 1. From the assumption, T_{ss} is a block diagonal matrix of which block elements are T_1,T_2,$\cdots$,T_C. v_j is all zero except n_j elements in the middle of v_j. The n_j elements are all positive and their indices correspond to those of T_j in T_{ss}. In the proof, the left eigenvector u_j of T_j will be substituted to this part.

Proof. From the Perron-Frobenius Theorem (Frobenius et al., 1912; Perron, 1907) that positive matrix has a real and positive eigenvalue with positive left and right eigenvectors, T_j, the block diagonal element of T_{ss}, has a positive left eigenvector u_j with eigenvalue λ_j for all $j = 1, 2, \cdots \mathrm{C}$. Then, as shown below, $v_j = \begin{pmatrix} 0 & 0 & \cdots & 0 & u_j^\top & 0 & \cdots & 0 \end{pmatrix}^\top$ where $n_1 + n_2 + \cdots + n_{j-1}$ zeros, u_j and $n_{j+1} + n_{j+2} + \cdots + n_{\mathrm{C}}$ zeros are concatenated, is a left eigenvector of T_{ss} with eigenvalue λ_j by the definition of eigenvector.

$$v_j^\top T_{ss} = \begin{pmatrix} \underbrace{0 \quad 0 \quad \cdots \quad 0}_{n_1+n_2+\cdots+n_{j-1}} & u_j^\top & \underbrace{0 \quad 0 \quad \cdots \quad 0}_{n_{j+1}+n_{j+2}+\cdots+n_{\mathrm{C}}} \end{pmatrix} \begin{pmatrix} T_1 & & & & \\ & \ddots & & & \\ & & T_j & & \\ & & & \ddots & \\ & & & & T_{\mathrm{C}} \end{pmatrix} \tag{9}$$

$$= \begin{pmatrix} 0 & 0 & \cdots & \lambda_j u_j^\top & 0 & 0 & \cdots & 0 \end{pmatrix} \tag{10}$$

$$= \lambda_j \begin{pmatrix} 0 & 0 & \cdots & 0 & u_j^\top & 0 & 0 & \cdots & 0 \end{pmatrix} \tag{11}$$

$$= \lambda_j v_j^\top \tag{12}$$

From the label propagation, we have,

$$\hat{y}^s = (I - T_{ss})^{-1} T_{st} \hat{y}^t. \tag{13}$$

By multiplying $v_j^\top (I - T_{ss})$ on the left and e_i on the right to the both sides in Equation 13 and combining with the assumption $\hat{y}^s = y^s$, we have, $\forall 1 \leq i \leq \mathrm{C}, i \neq j$,

$$v_j^\top T_{st} \hat{y}^t e_i = v_j^\top (I - T_{ss}) \hat{y}^s e_i \tag{14}$$
$$= v_j^\top (I - T_{ss}) y^s e_i \tag{15}$$
$$= (1 - \lambda_j) v_j^\top y^s e_i \tag{16}$$
$$= 0 \tag{17}$$

The last zero comes from the definition of v_j. □

A.1 Extension of Theorem 1

In this subsection, we offer the modified version of Thm. 1 when the source features are slightly perturbed from the ideally clustered condition and the other assumption $y^s = \hat{y}^s$ holds. We start from representing T_{ss} as follows to indicate the perturbation.

$$T_{ss} = T_{ss}^{(0)} + \delta T_{ss} \tag{18}$$

where, δT_{ss} is assumed to be sufficiently small under infinite norm and $T_{ss}^{(0)}$ is a block diagonal transition matrix when the source features are ideally clustered as stated in Thm. 1.

In the proof above, we showed eigenvalue λ_j and its corresponding eigenvector v_j of $T_{ss}^{(0)}$ of which j^{th} block elements are positive and the others are all zero. We denote those λ_j and v_j by $\lambda_j^{(0)}$ and $v_j^{(0)}$. According to perturbation theory of eigenvalue and eigenvector (Greenbaum et al., 2019), the eigenvector can be approximated by first order when the perturbation is small.

$$v_j = v_j^{(0)} + O(||\delta T_{ss}||_\infty) \tag{19}$$
$$\lambda_j = \lambda_j^{(0)} + O(||\delta T_{ss}||_\infty). \tag{20}$$

More generally and precisely,

$$||v_j - v_j^{(0)}|| \leq m_j ||v_j^{(0)}|| ||\delta T_{ss}|| \tag{21}$$

where, the norm is vector or matrix 2-norm and m_j is determined by $T_{ss}^{(0)}$. For the sake of simplicity, we use Big-O notation in Eq. 19 and Eq. 20.

Now, we reuse Eq. 16 from the proof of Theorem 1 since it is still valid under the modified condition.

$$v_j^T T_{st} \hat{y}^t e_i = (1 - \lambda_j) v_j^T y^s e_i. \tag{22}$$

We apply Eq. 19 to the right hand side as follows,

$$(1 - \lambda_j) v_j^T y^s e_i = (1 - \lambda_j)(v_j^{(0)} + O(||\delta T_{ss}||_\infty))^T y^s e_i \tag{23}$$
$$= O(||\delta T_{ss}||_\infty) \tag{24}$$

where $i \neq j$. Eq. 24 holds because only j^{th} block elements of $v_j^{(0)}$ are nonzero. We also used the fact that y^s is bounded by 0 and 1. Similarly, the left hand side of Eq. 22 can be transformed as follows,

$$v_j^T T_{st} \hat{y}^t e_i = (v_j^{(0)} + O(||\delta T_{ss}||_\infty))^T T_{st} \hat{y}^t e_i \tag{25}$$

$$= v_j^{(0)T} T_{st} \hat{y}^t e_i + O(||\delta T_{ss}||_\infty). \tag{26}$$

The second term of Eq. 26 holds because T_{st} and $\hat{y}^t$ are bounded by 0 and 1. Finally, by combining Eq. 24 and Eq. 26, we have,

$$v_j^{(0)T} T_{st} \hat{y}^t e_i = O(||\delta T_{ss}||_\infty). \tag{27}$$

Eq. 27 implies that if the perturbation is sufficiently small *i.e.*, $||\delta T_{ss}||_\infty << 1$ and a target data is enough to be predicted as i^{th} class through label propagation, then the transitions from source data of all but i^{th} class to the target data is negligible because $v_j^{(0)}$ is positive for j^{th} block and zero for others. It is the same with the conclusion of Theorem 1. In addition, the more strongly the target data is classified as i^{th} class *i.e.*, the corresponding element of $\hat{y}^t$ becomes greater, the smaller the transitions from source data in the other classes are, indicating the segregation against the other classes.

Practically, the coefficients for L_{cycle} and L_{cycle} are scheduled to facilitate the clustering of source features correctly at the early stage of training. Thus we may assume that T_{ss} is marginally perturbed around the ideally clustered one when our cycle loss takes effect.

B Experimental Detail

B.1 Training Detail

Scheduling the effect of losses To reduce the effect of noisy signal from L_{dann} and L_{cycle} during the early stages of training, a weight balance factor $\lambda = \frac{2}{1+exp(-\gamma \cdot p)} - 1$ is applied in Eq. 6. A constant γ determines the rate of increase of λ; p is the progress of training, which proceeds from 0 to 1. The parameter was introduced (Ganin et al., 2016) to make a classifier less sensitive to the erroneous signals from the discriminator in the beginning. Throughout the experiments, γ was set to 10.

Hyperparameter Although it would be ideal to avoid utilizing labels from the target domain in the hyperparameter optimization, it seems that no globally acceptable method exists for this. One possibility (Ganin et al., 2016) is reverse validation scheme but this may not be accurate enough to estimate test accuracy (Bousmalis et al., 2016). In addition (Bousmalis et al., 2016), applications exist where the labeled target domain data is available at the test phase but not at the training phase. Hence, we adopted the protocol of (Bousmalis et al., 2016) that exploits a small set of labeled target domain data as a validation set; 256 samples for the Amazon review experiment and 1,000 samples for the other experiments (Bousmalis et al., 2016; 2017; Saito et al., 2017). During training, Adam optimizer (Kingma & Ba, 2015) with learning rate of 10^{-3} was utilized. Exponential moving averaging was applied to the optimization trajectory.

Batch Sizes It is an inherent characteristic of our method that each data sample affects the graph structure. So it is important for each class sample in each batch to represent its classes accurately. In other words, the transition matrix can be corrupted by biases in the samples. Therefore, the number of data samples in each class in a batch should be sufficient to avoid any likely bias. To address this problem, we performed experiments with batch size of up to 384 and observed very little improvement beyond a batch size of 128. So we fixed the batch size to 128 for Digit & Object dataset. For the ImageCLEF-DA dataset, we set the batch size to 36 because of limited computing resource.

B.2 Adaptation Settings

MNIST → MNIST-M The MNIST database of hand-written digits (LeCun et al., 1998) consists of digit images with 10 classes and MNIST-M (Ganin et al., 2016) consists of MNIST digits blended with natural color patches from the BSDS500 dataset (Arbelaez et al., 2011). In addition, following

other work (Pinheiro, 2018) the colors of the MNIST images were inverted randomly, because their colors are always white on black, whereas the MNIST-M images exhibit various colors.

MNIST ↔ USPS USPS (Denker et al., 1989) is another dataset of hand-written images of digits, with 10 classes. USPS contains 16×16 images and the size of the USPS image is upscaled to 28×28, which is the size of the MNIST image in our experiment. The evaluation protocol of CYCADA (Hoffman et al., 2018) is adopted.

SVHN → MNIST The Street View House Numbers (SVHN) (Netzer et al., 2011) dataset consists of images of house numbers acquired by Google Street View. The natural images that it contains, are substantially different from the line drawings in the MNIST dataset. The size of each MNIST image is upscaled to 32×32, which is the size of SVHN images.

SYN DIGITS → SVHN SYN DIGITS dataset is synthetic number dataset which is similar to the SVHN dataset (Ganin et al., 2016). The most significant difference between the SYN DIGITS dataset and the SVHN dataset is the untidiness (Ganin et al., 2016) in the background of real images.

CIFAR ↔ STL Both CIFAR dataset (Krizhevsky & Hinton, 2009) and STL dataset (Coates et al., 2011) are 10-class datasets that contain images of animals and vehicles. Not overlapped classes are removed to make a 9-class domain adaptation task (Shu et al., 2018). We used the larger network only for this experiment.

ImageCLEF-DA[2] The twelve common classes of three publicly available dataset (*Caltech-256*, *ImageNet ILSVRC2012*, and *PASCAL VOC2012*) are selected to form visual domain adaptation tasks. We perform all six possible adaptation scenarios among these three dataset.

C HYPERPARAMETERS

We searched hyperparameters within $\alpha = \{0, 0.01, 0.1, 1\}$, $\beta = \{0.01, 0.1, 1\}$, $\mu = \{0, 0.01\}$ and $\eta = \{0, 0.1\}$. Perturbation to the LGAN generator, *i.e.* z, is fixed to 0.5 for all experiments. The best hyperparameters for each task is shown in Table. 3.

Table 3: Hyperparameters for each task

Task	α	β	μ	η
MNIST → MNIST-M	0.1	1	0.01	0
MNIST → USPS	0.1	1	0.01	0
USPS → MNIST	0.01	1	0.01	0.1
SVHN → MNIST	0	1	0.01	0
DIGITS → SVHN	0.1	1	0.01	0.1
CIFAR → STL	0.1	1	0	0.1
STL → CIFAR	0.1	0.1	0.01	0.1
I → P	0.01	0.1	0	0.1
P → I	0.01	1	0	0.1
I → C	0.1	0.01	0	0.1
C → I	0.1	0.1	0.01	0.1
C → P	0.01	0.1	0	0.1
P → C	0	1	0.01	0

D ABLATION STUDY

D.1 SCALE PARAMETER

Setting an appropriate value for the scale parameter, σ, is important because it has a substantial role in determining the transition matrix, T. Therefore, we conducted several experiments with fixing σ to various values. For these experiments, we excluded L_{local} to observe the effect of σ. 'Adapt' means that the σ is learned to adapt according to the embedded features.

[2]https://www.imageclef.org/2014/adaptation

Table 4: Accuracy (%) on Digit dataset with different σ values. All experiments were run 5 times.

Source Target	MNIST MNIST-M	MNIST USPS	USPS MNIST	SVHN MNIST	DIGITS SVHN
0.1	94.8±0.4	96.5±0.3	98.3±0.1	76.7±7.6	90.6±0.5
1	96.4±0.5	97.2±0.1	95.8±6.7	98.2±0.1	92.0±0.2
10	84.8±3.5	95.4±0.5	95.8±0.3	74.2±4.9	87.2±0.4
Adapt	96.5±0.1	97.3±0.2	98.6±0.1	98.2±0.2	92.1±0.2

For four out of five scenarios, fixing σ to 1 performed better than fixing it to 0.1 or 10. With this observation, we initialized σ to 1 took it as a trainable variable. The result of adaptively learning σ is reported at the bottom row of the table. Compared to fixing σ to 1, adaptively learning σ achieved better accuracy and had a lower standard deviation range which means that it is more stable. We also would like to highlight that our model is robust to the initial value of σ. We conducted extensive experiments with initializing σ to 0.1, 1 and 10 and taking it as a trainable variable.

Table 5: Accuracy (%) on Digit dataset with different σ initializations. All experiments were run 5 times.

Source Target	MNIST MNIST-M	MNIST USPS	USPS MNIST	SVHN MNIST	DIGITS SVHN
0.1	96.4±0.2	97.1±0.1	98.6±0.1	98.0±0.5	91.9±0.3
1	96.5±0.1	97.3±0.2	98.6±0.1	98.2±0.2	92.1±0.2
10	96.3±0.1	96.9±0.2	98.6±0.2	91.9±8.8	92.1±0.1

Except for SVHN $\rightarrow$ MNIST transfer task with setting initial σ value to 10, the initial value of σ has a minute influence to the accuracy. We believe that adaptively learning the scale parameter can be usefully employed in any other graph-based method. The learned σ values for various scenarios are as follows.

Table 6: Accuracy of the learned σ on Digit dataset with different σ initializations.

Source Target	MNIST MNIST-M	MNIST USPS	USPS MNIST	SVHN MNIST	DIGITS SVHN
0.1	1.20±0.44	1.04±0.13	1.15±0.54	1.18±0.10	1.26±0.12
1	1.29±0.13	1.00±0.40	1.07±0.46	1.17±0.06	1.26±0.13
10	1.30±0.14	0.96±0.11	1.13±0.42	1.21±0.07	1.29±0.10

It seems that σ adaptively learns its value according to the transfer task, regardless of its initialization.

D.2 Loss function for cycle consistency

We tried l_2 loss and cross entropy loss to enforce cycle consistency as well. We excluded L_{local} to compare the effectiveness of these functions.

Table 7: Accuracy (%) on Digit dataset with different loss functions for cycle consistency. All experiments were run 5 times. CE refers to the cross entropy.

Source Target	MNIST MNIST-M	MNIST USPS	USPS MNIST	SVHN MNIST	DIGITS SVHN
l_1 loss	96.5±0.1	97.3±0.2	98.6±0.1	98.2±0.2	92.1±0.2
l_2 loss	96.4±0.2	97.1±0.2	98.3±0.3	98.0±0.1	92.0±0.2
CE loss	96.3±0.3	96.9±0.2	98.5±0.2	96.6±2.9	91.8±0.2

For all Digit dataset adaptation experiments, evaluating cycle consistency with l_1 norm achieved the highest accuracy. We speculate that l_1 norm is more numerically stable or provides more effective gradients than other functions in this case.

E NON VISUAL DATASET EXPERIMENT

The Amazon Reviews (Blitzer et al., 2007) dataset provides a non-visual domain for domain adaptation experiments. It contains reviews of books, DVDs, electronics, and kitchen appliances encoded as 5,000-dimensional feature vectors containing unigrams and bigrams of the texts with binary labels. Four- and five-star reviews are labeled 'positive'; reviews with fewer stars are labeled 'negative'. We used 2,000 labeled source data and 2,000 unlabeled target data for training, and between 3,000 to 6,000 target data for testing.

Tab. 8 shows that our method performs better than DANN (Ganin et al., 2016), VFAE (Louizos et al., 2016) and ATT (Saito et al., 2017) on the Amazon Reviews data in six out of twelve experiments. Our method was more accurate than DANN in nine out of twelve settings, showing approximately 2.0% higher classification accuracy on average.

Table 8: Accuracy (%) for nonvisual domain adaptation with Amazon Reviews dataset

Source Target	book dvd	book elec	book kit	dvd book	dvd elec	dvd kit	elec book	elec dvd	elec kit	kit book	kit dvd	kit elec
VFAE (Louizos et al., 2016)	79.9	79.2	81.6	75.5	78.6	82.2	72.7	**76.5**	85.0	72.0	73.3	83.8
DANN (Ganin et al., 2016)	78.4	73.3	77.9	72.3	75.4	78.3	71.1	73.8	85.4	70.9	74.0	84.3
ATT (Saito et al., 2017)	80.7	**79.8**	**82.5**	73.2	77.0	**82.5**	**73.2**	72.9	86.9	**72.5**	**74.9**	84.6
Ours	**81.3**	78.3	79.7	**77.2**	**79.0**	**82.5**	70.8	73.3	**87.1**	71.8	73.5	**85.4**
(std)	±0.0	±0.2	±0.5	±1.6	±0.7	±0.4	±0.3	±1.2	±0.2	±0.7	±0.8	±0.1

book: books, dvd: DVDs, elec: electronics, kit: kitchen appliances