Unsupervised Multi-source Domain Adaptation with Graph Convolution Network and Multi-alignment in Mixed Latent Space

Hongqing Zhu (hqzhu@ecust.edu.cn)
East China University of Science and Technology
Dong Chen
East China University of Science and Technology
Suyi Yang
King's College London
Yiwen Dai
East China University of Science and Technology

Research Article

Keywords: Multi-source domain adaptation , Latent space , Graph Convolutional Network , Moment matching , Image classification

Posted Date: April 8th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1528207/v1

Signal, Image and Video Processing manuscript No.
(will be inserted by the editor)

Unsupervised Multi-source Domain Adaptation with Graph Convolution Network and Multi-alignment in Mixed Latent Space

Dong Chen[1] · **Hongqing Zhu**[1]* · **Suyi Yang**[2] · **Yiwen Dai**[1]

Received: date / Accepted: date

Abstract This paper proposes an unsupervised Multi-source Domain Adaptation algorithm with Graph convolution network and Multi-alignment in mixed latent space (MDA-GM), which leverages domain labels, data structure, and category labels in a unified network but improve domain-invariant semantic representation by several innovations. Specifically, a novel data structure alignment is proposed to exploit the inherent properties of different domains while using current domain alignment and classification result alignment. Through this design, category consistency can be considered in both latent space, and domain and structure discrepancy between different source domains and the target domain can be eliminated. Moreover, we also use category alignment based on both CNN and GCN features to optimize category decision boundary. Experiment results show that the proposed method brings sufficient improvement especially for adaptation tasks with large shift in data distribution.

Keywords Multi-source domain adaptation · Latent space · Graph Convolutional Network · Moment matching · Image classification

1 Introduction

Deep learning algorithms assume that the features of training set and test set obey the same distribution, which means models trained on large-scale annotated training data cannot be reused on other domains. Domain adaptation mainly solves this problem by establishing knowledge transfer from labeled source domain to unlabeled target domain [1]. Single-source Domain Adaptation (SDA) has been frequently researched but in most practical scenarios, available annotation data may come from multiple domains with distributions different from the target and even differ from each other. In this case, the SDA method can be simply applied by combining multiple source domains into single-source domain, which is call source-combined domain adaptation. Source-combined domain adaptation has made a progress as training data has been enriched. However, data bias still exists between each source domain and target domain as well as between different source domains. Forced transfer from domains with low correlation will suppress the performance of the target model, resulting in the problem of "negative transfer". In order to avoid these problems, Multi-source Domain Adaptation (MDA) is researched. Recent mainstream MDA methods either use different feature transformation methods to achieve feature alignment in latent spaces [2,3], or build intermediate domains through generator or encoder [4].

Graph Convolutional Adversarial Network (GCAN) for Unsupervised Domain Adaptation (UDA) [5] verifies that data structure, domain labels, and category labels play an important role in connecting labeled source domain and unlabeled target domain. Although this is discovered on SDA tasks, these three types of information may also be contributing to analysis on different source domains. Another inspiring phenomenon might be that researches processing non-Euclidean space data (eg. for relational reasoning, knowledge graph) all exploit structure information to certain extent, which indicates its potential importance. Data structure generally reflects the inherent attributes of dataset, such as data distribution, geometric data structure, etc. However, current MDA studies have not utilized structure information from the perspective of latent space transformation. Recently, some methods adopt domain alignment [3, 6] where do-

Hongqing Zhu*
E-mail: hqzhu@ecust.edu.cn
[1] School of Information Science and Engineering, East China University of Science and Technology. Shanghai 200237, China
[2] Department of Mathematics, Natural, Mathematical & Engineering Sciences, King's College London, Strand, London WC2R 2LS, United Kingdom
This work was supported by the National Nature Science Foundation of China under Grant 61872143.

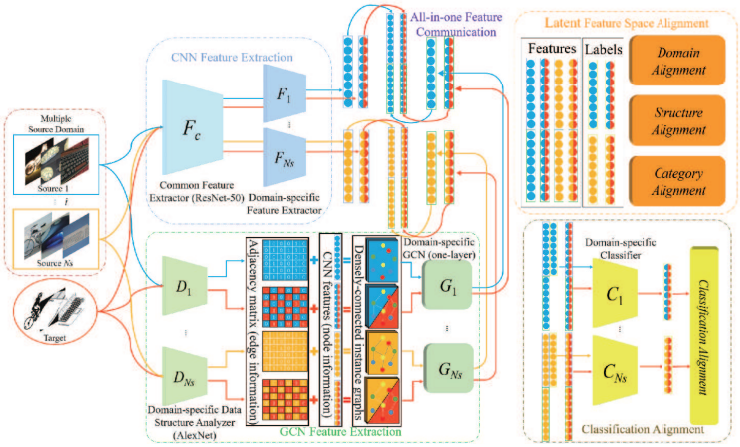


Fig. 1: The architecture of the proposed framework. The network is constituted of four modules: CNN feature extraction module, GCN feature extraction module, latent feature space alignment module, and classification alignment module.

main label represents global features of different domains. This strategy would help the algorithm discriminate global distribution of source domain and target domain. Category labels, especially pseudo labels of target data, are also used to strengthen semantic alignment and optimize decision boundary of classification. In conclusion, adopting these three types of information has been verified effective in reducing domain shift from many aspects.

In this paper, we propose an end-to-end unsupervised Multi-source Domain Adaptation algorithm with Graph convolution network and Multi-alignment in mixed latent space (MDA-GM). This algorithm continues to leverage domain labels, category labels, data structure information and seek to minimize distribution discrepancy. Except for the classification result alignment conducted during classification stage, the other three alignments are implemented in the stage of feature extraction. We map each pair of source domain and target domain data into a domain-specific latent feature space. Two individual feature extraction modules are designed to construct Convolutional Neural Network (CNN)-based and GCN-based latent feature spaces. Domain alignment and structure alignment are realized by aligning mapped features in hybrid latent feature spaces composed of the CNN-based and GCN-based latent feature spaces. The category alignment is designed using the real label of source domain and the pseudo-label of target domain. With the classification result alignment, the classification bias for the same target data between each classifier will be reduced. By these alignment strategies in multiple latent spaces, the proposed network can achieve good domain adaptation performance.

Main contributions of this work include: (i)we establish an end-to-end unsupervised MDA with GCN and multi-alignment in mixed latent feature space; (ii)we propose a data structure alignment scheme to ensure data with the same structure can be more closely clustered; (iii)an all-in-one feature communication step is set before feature alignment so that domain labels and data structure information can be mutually constrained; (iv)the proposed category alignment focuses on local features of categories, which together with category centroids, further optimizes category decision boundary.

2 Methodology

In this section, we will elaborate the principle of the four main parts in this method: CNN feature extraction, GCN feature extraction, latent feature space alignment, and classification alignment as shown in Fig. 1. Fig. 2 visualizes the latent feature space alignment module in detail to examplify the varied distributions by each alignment individually. In practice, domain, structure and category alignment conduct simultaneously. Distribution before and after alignments of the source domain $i(i=1)$ (upper) and source domain N_S (lower) with the target domain are also presented in Fig. 2.

1) Problem setting and notations. There are N_S different source domains and their underlying source distributions are defined as $\{p_{S_i}(x,y)\}_{i=1}^{N_S}$. The labeled source domain images $\{(X_{S_i},Y_{S_i})\}_{i=1}^{N_S}$ are drawn from those distributions respectively, where $X_{S_i}=\{x_{s_i}^j\}_{j=1}^{|X_{S_i}|}$ denotes source images and $Y_{S_i}=\{y_{s_i}^j\}_{j=1}^{|Y_{S_i}|}$ denotes the corresponding labels. Distribution of the target domain T is defined as $p_T(x,y)$, and the unlabeled target domain images are defined as $X_T=\{x_T^j\}_{j=1}^{|X_T|}$. The proposed framework studies the MDA of closed set, so there is no category bias between each domain. We use mini-batch $\hat{X}=\{(\hat{X}_{S_1},\hat{Y}_{S_1}),(\hat{X}_{S_2},\hat{Y}_{S_2}),\ldots,(\hat{X}_{S_{N_S}},\hat{Y}_{S_{N_S}}),\hat{X}_T\}$ to describe the source data with labels and target data, where $\hat{X}$ represents one training batch.

2) CNN feature extraction module. CNN Feature Extraction module consists of common feature extractors F_c and domain-specific feature extractors $\{F_i\}_{i=1}^{N_S}$. The labeled

source domain data and unlabeled target domain data are sent to CNN feature extraction module. Different from previous MDA that merely use one common feature extractor, we design a comprehensive feature extraction network that extracts domain-invariant features for each pair of source and target domains respectively while concerning domain-specific decision boundaries. This feature extraction module firstly uses a common feature extractor when following by N_S domain-specific feature extractors. Both source and target domain data will pass through a common feature extractor $F_c(\hat{X})$. Then, different source domain data pass through their own domain-specific feature extractor.

$$\{\mathrm{CF}_{S_i}(\hat{X}_{S_i})=F_i(F_c(\hat{X}_{S_i}))\}_{i=1}^{N_S}, \tag{1}$$

where $\mathrm{CF}_{S_i}(\hat{X}_{S_i})$ denotes the CNN features of source domain i. The target domain data will go through each domain-specific feature extractor.

$$\{\mathrm{CF}_T^{S_i}(\hat{X}_T)=F_i(F_c(\hat{X}_T))\}_{i=1}^{N_S}, \tag{2}$$

where $\mathrm{CF}_T^{S_i}(\hat{X}_T)$ denotes CNN features of target domain with respect to source domain i. This additional architecture largely retains domain-specific information between different source domains. Meanwhile, source and target domain distributions are aligned with less computational complexity.

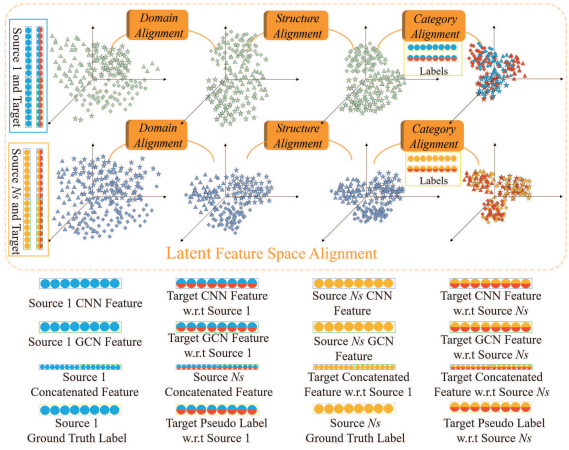


Fig. 2: A discrete visualization of the latent feature space by each individual alignment. Different categories are indicated in points with different shapes and different domains are represented by different colors.

3) GCN feature extraction module. GCN Feature Extraction module consists of domain-specific data structure analysers $\{D_i\}_{i=1}^{N_S}$ and domain-specific GCNs $\{G_i\}_{i=1}^{N_S}$. Current research in the field of MDA has not constructed graph-based latent feature space for structure alignment. In fact, reconstruction-based domain adaptation methods focus on modelling data structure information and have obtain impressive performance, which to some extent verifies its contribution in domain adaptation. In [5], GCN has also been proved effective in processing graph data that contains strong structure information. GCN is a graph embedding technology used to obtain feature representation $h_v \in \mathbb{R}^m$ of graph node v, where h_v is an m-dimension vector and can be used to produce an output o_v based on the node label. In our method, the input image can be viewed as a graph node v, and image classification is implemented based on node classification.

Given that an undirected graph $\bar{G}$ has $\tilde{A}=A+I$ where A represents the adjacency matrix and I denotes the identity matrix, the degree matrix can be calculated as $\tilde{D}_{ii}=\sum_j \tilde{A}_{ij}$. The parameter matrix and activation matrix (input node matrix) of the l^{th} layer are denoted as $W^{(l)} \in \mathbb{R}^{C\times F}$ and $H^{(l)} \in \mathbb{R}^{N\times C}$ respectively. When l^{th}=0, H is input node matrix. Based on above settings, the graph convolution model can be expressed by

$$H^{(l+1)}=\sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}), \tag{3}$$

where σ represents the activation layer function. To align structure features, we use GCN to process indirect instance graphs and extract structure features of different domain data. In our framework, node feature and the relationship between nodes are the feature of input image data and their mutual relationship. To obtain adjacency matrix A, we firstly input source domain and target domain data pairs $\{(\hat{X}_{S_i},\hat{X}_T)\}_{i=1}^{N_s}$ into domain-specific data structure analyzers (AlexNet) to obtain structure features of source domain and target domain using

$$\{\mathrm{SF}_{S_i}(\hat{X}_{S_i})=D_i(\hat{X}_{S_i})\}_{i=1}^{N_S}, \{\mathrm{SF}_T^{S_i}(\hat{X}_T)=D_i(\hat{X}_T)\}_{i=1}^{N_S}. \tag{4}$$

Using these structure features, the adjacency matrix A of source domain and target domain can be obtained by

$$\begin{aligned}\{A_{S_i}&=\mathrm{SF}_{S_i}(\hat{X}_{S_i})\mathrm{SF}_{S_i}(\hat{X}_{S_i})^{\mathrm{T}}\}_{i=1}^{N_S},\\ \{A_T^{S_i}&=\mathrm{SF}_T^{S_i}(\hat{X}_T)\mathrm{SF}_T^{S_i}(\hat{X}_T)^{\mathrm{T}}\}_{i=1}^{N_S},\end{aligned} \tag{5}$$

respectively. With the calculated adjacency matrix and the learned CNN features, dense-connected instance graphs for GCN can be constructed. Then, they are delivered to domain-specific GCN which maps densely-connected instance graphs to GCN-based latent feature space. The obtained GCN features of source domain i and target domain with respect to source domain i are denoted as $\{\mathrm{GF}_{S_i}(\hat{X}_{S_i})=G_i(\mathrm{CF}_{S_i}(\hat{X}_{S_i}),A_{S_i})\}_{i=1}^{N_S}$ and $\{\mathrm{GF}_T^{S_i}(\hat{X}_T)=G_i(\mathrm{CF}_T^{S_i}(\hat{X}_T),A_T^{S_i})\}_{i=1}^{N_S}$ respectively.

4) Latent feature space alignment module. In this paper, the Maximum Mean Discrepancy (MMD) is implemented to measure the discrepancy of latent spaces from different domains. It is formulated in Reproducing Kernel Hilbert Space $\mathscr{H}$ as follows:

$$MMD(X_S,X_T)=\left\|\frac{1}{|X_S|}\sum\phi(x_S^i)-\frac{1}{|X_T|}\sum\phi(x_T^i)\right\|_{\mathscr{H}}^2, \tag{6}$$

where $\phi(\cdot)$ denotes a mapping function from the original samples space to $\mathscr{H}$ space. By reducing the discrepancy in CNN-based latent feature space, domain alignment can be realized.

$$\mathscr{L}_D = \frac{1}{N_S}\sum_{i=1}^{N_S} MMD(\mathrm{CF}_{S_i}(\hat{X}_{S_i}), \mathrm{CF}_T^{S_i}(\hat{X}_T)) \tag{7}$$

Reducing the discrepancy in GCN-based latent feature space can reduce structure discrepancy between different source domains and target domains.

$$\mathscr{L}_S = \frac{1}{N_S}\sum_{i=1}^{N_S} MMD(\mathrm{GF}_{S_i}(\hat{X}_{S_i}), \mathrm{GF}_t^{S_i}(\hat{X}_T)). \tag{8}$$

We set an simple but effective all-in-one feature communication step before feature alignment. Specifically, we concatenate CNN feature and GCN feature that domain labels and data structure information can be mutually constrained and enable domain alignment and structure alignment at the same time.

$$\mathscr{L}_M = \frac{1}{N_S}\sum_{i=1}^{N_S} MMD(\mathrm{F}_{S_i}(\hat{X}_{S_i}), \mathrm{F}_T^{S_i}(\hat{X}_T)) \tag{9}$$

$\mathrm{F}_{S_i}(\hat{X}_{S_i})$ and $\mathrm{F}_T^{S_i}(\hat{X}_T)$ denote concatenated feature.

Although the proposed framework firstly considers both domain invariance and structure consistency, some incorrectly adapted samples that meet both conditions would still fail to be classified due to lack of category-discriminative features. Therefore, this paper additionally designs category alignment in the training process. Each domain-specific classifier is trained using the corresponding source domain data and labels, and the corresponding classification loss is

$$\mathscr{L}_{cls}(\hat{X}_{S_i}, \hat{Y}_{S_i}) = \mathbb{E}[J(C_i(\mathrm{F}_{S_i}(\hat{X}_{S_i})), \hat{Y}_{S_i})], \tag{10}$$

where J is the cross entropy loss, $\hat{Y}_{S_i}$ denotes Ground truth label of source domain i. Since there is no target domain label available, we use domain-specific classifiers to obtain pseudo-labels $\hat{Y}_t^{S_i}$ for target domain using

$$\hat{Y}_T^{S_i} = \mathrm{soft\,max}\{C_i(\mathrm{F}_T^{S_i}(\hat{X}_T))\}. \tag{11}$$

In this paper, we choose "centroid-to-centroid distance" to accomplish category alignment. When we calculate the centroid for each category, all pseudo-labeled (correct or wrong) samples are used together, and the negative effects brought by wrong samples are neutralized by correct samples to some extent. Here, the centroid of samples of the same category can be defined by

$$CT = \sum \bar{f}(x_i^{N_k})/N_k, \tag{12}$$

where N_k represents the number of samples, $x_i^{N_k}$ denotes all samples belonging to category k, and $\bar{f}(\cdot)$ denotes corresponding feature. Meanwhile, we need to align all same categories between each pair of source domain and target domain according to the theoretical analysis in [7]. But it would lead to enormous amount of calculation. In fact, category decision boundary is mainly determined by the nearest neighboring data samples of different categories. Therefore, in actual category alignment, we reduce the distance of sample centroid of the same categories. For the sample centroid of different categories, we only increase discrepancy between the nearest different centroid. The category alignment could be reformulated as

$$\begin{aligned}\mathscr{L}_{cat} &= \frac{1}{C}\sum_{r=1}^{C} MMD(CT_{S_i}^r, CT_T^r) \\ &- (\frac{\lambda_C}{2}(MMD(CT_{S_i}^{r_{s1}}, CT_{S_i}^{r_{s2}}) + MMD(CT_T^{r_{t1}}, CT_T^{r_{t2}}))),\end{aligned} \tag{13}$$

r_{s1}, r_{s2} represent the closest sample centroid of different categories of source domain, r_{t1}, r_{t2} represent the closest sample centroid of different categories of target domain, and λ_C is set to 0.01 according to [7].

5) Classification alignment module. Classification Alignment module is composed of domain-specific classifiers $\{C_i\}_{i=1}^{N_S}$. It ensures that the classification results of N_S classifiers on target domain maintain consistent. The consistency of classification result loss between domain-specific classifiers could be reformulated as follows:

$$\begin{aligned}\mathscr{L}_{coc} &= 2/(N_S(N_S-1)) \\ &\times \sum_{n_1=1}^{N_S-1}\sum_{n_2=n_1+1}^{N_S} |\, C_{n_1}(\mathrm{F}_t^{S_{n1}}(\hat{X}_T)) - C_{n_2}(\mathrm{F}_t^{S_{n2}}(\hat{X}_T))\,|.\end{aligned} \tag{14}$$

Finally, the overall loss in training process is defined as

$$\mathscr{L}_{all} = \mathscr{L}_{cls} + \alpha\mathscr{L}_D + \beta\mathscr{L}_{cat} + \gamma\mathscr{L}_{coc} + \zeta\mathscr{L}_S, \tag{15}$$

where α, β, γ, ζ is the trade-off parameters. The target domain data are input to the trained feature extraction module, and then passes through trained domain-specific classifiers to generate classification results.

3 Experiments

1) Datasets. Office-31 dataset [10] is an unbalanced dataset containing 4110 images in 31 categories: Amazon (A), Webcam (W) and DSLR (D). Set A contains 2817 images, set W collects 498 images taken by webcam and set D collects 795 images taken by digital SLR cameras. We use all domain combinations and establish three MDA tasks: A, W→D; W, D→A; A, D→W. Office-Home dataset contains 15,500 images of 65 categories taken in the offices and homes [11]. It consists of 2427 (Art) images, 4365 (Clipart) images, 4439 (Product) images, and 4269 (Real world) images. We use all domain combinations and establish four MDA tasks: C, P, R→A; A, P, R→C; A, C, R→P; A, C, P→R.

2) Implementation details. All experiments run on PyTorch 1.5.0 by a server with Intel (R) Core (TM) i7-10700K

CPU and NVIDIA GeForce RTX 2080 Ti GPU. The method is trained for 10000 epochs on Office-31, and 15000 epochs on Office-Home. All training images are randomly cropped into 224×224 patches as input. We use the batch stochastic gradient descent method with momentum of 0.9 and batch of 32 to finetune all trainable parameters. The learning rate follows the rule in [20], that is $\eta=\eta_0/(1+10p)^{0.75}$, η_0=0.01. The pre-trained ResNet-50 serves as a common feature extractor. We use the AlexNet model pre-trained on ImageNet as domain-specific structure analyzer to extract structure feature with 1000 dimensional output. Domain-specific GCNs are designed as single-layer GCN networks according to the framework of GCAN [5].

3) Discussion of trade-off parameters. This experiment explores the influence of trade-off parameters in (15), where γ, ζ control the contribution of category alignment and structure alignment. The classification accuracy under different values of γ and ζ of task D,W→A is shown in Table 1, we observe that it is best to choose the dynamic adjustment strategy for γ, ζ. It is because in the early stage of training, there will be the influence of noise signal. It is more reasonable to adopt the strategy of dynamic adjustment instead of fixed adaptive factor. For other two parameters α, β, we follow the settings in MFSAN [3] $\alpha = \beta = (2/(1+\exp(-10p)))-1$, p is changes linearly from 0 to 1. Considering that both domain alignment and structure alignment are achieved by calculating MMD, and γ and ζ are set the same. we propose an all-in-one feature communication step, through this step domain alignment and structure alignment can be complied at one time. The results show that using this dynamic strategy not only improves the computing efficiency, but also improves the adaptation performance.

Table 1: Classification accuracy (%) of task D,W→A on Office-31 with different γ and ζ, "AFC" denotes all-in-one feature communication and "Dynamic" denotes $(2/(1+\exp(-10p)))-1$, p is changes linearly from 0 to 1.

Adaptation factor γ	Adaptation factor ζ	AFC	D,W→A
0.2	0.0	✗	72.9
0.4	0.0	✗	74.2
0.6	0.0	✗	73.3
0.8	0.0	✗	73.1
1.0	0.0	✗	72.5
Dynamic	0.2	✗	78.8
Dynamic	0.4	✗	79.9
Dynamic	0.6	✗	78.4
Dynamic	0.8	✗	78.9
Dynamic	1.0	✗	77.5
Dynamic	Dynamic	✗	84.9
Dynamic	Dynamic	✓	86.1

4) Convergence analysis. We visualized the training process of D, W→A task on Office-31 and R, C, A→P task on Office-Home shown in Fig. 3. It can be seen that the training loss of each source domain decreases stably as number of training epochs increases, which indicates that MMD loss, classification consistency loss and cross entropy loss all converge gradually. Due to the limitation of GPU, we used the training strategy of sending each source domain into the network one by one (the feeding order of D, W→A task and R, C, A→P task is D, W and R, C, A) for training. Source domain data sent into the network to align with the target in a later sequence would see a slower decline of its loss. However, in later period of the training, all losses gathered at the same range and the accuracy of the target domain gradually increases to a stable level.

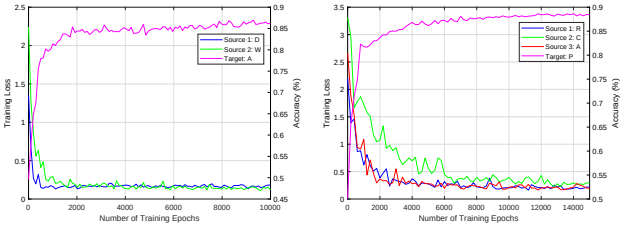


Fig. 3: Convergence analysis of MDA-GM: the trend of training loss and classification accuracy in the training process of D, W→A task (left) and R, C, A→P task (right).

5) Comparison with state-of-art. We choose several advanced MDA methods as the baseline: DCTN [2], M^{3}SDA [6], MFSAN [3], MSCLDA [7], SImpAl [12], MDADFRE [13], MDAN [14], MDMN [15], DARN [16], CMSDA [17]. In addition, some SDA methods are also compared, including: ResNet [18], DDC [8], DAN [9], D-CORAL [19], RevGrad [20], RTN [21], GCAN [5], FixBi [22]. In detail, three standards for MDA are adopted: "Single Best" refers to the best performance in SDA which evaluates whether the best result can be further improved by adding other source domains; "Source Combine" refers to the performance obtained by SDA after all source domains are merged into one domain; "Multi Source" represents MDA performance. Table. 2 and Table. 3 show the experimental results of our MDA-GM in comparison with other algorithms. In specific, MDA-GM brings sufficient improvement to existing domain adaptation methods. The average accuracy rate on Office-31 and Office-Home is 95% and 80.1% respectively, which achieves the highest accuracy on most adaptation tasks. Compared with baseline MFSAN, category alignment for CNN and GCN features, and structure alignment in GCN-based feature space we introduced brings a significant improvement. For instance, average accuracy on Office-31 exceeds baseline by 4.8%, and increases even up to 13.4% on D, W→A task which suf-

fers from large domain shift. The average accuracy rate on the Office-Home exceeds baseline by 6%. FixBi achieves the second-best results using advanced bidirectional matching and self-penalization techniques. In general, classification results by the proposed method demonstrate the effectiveness of introducing GCN-based latent feature space with structure alignment and additional category alignment. The domain adaptation with multiple source domains generally shows better results than single best and source-combine results. This shows that enriching source domain data can further improve classification results. Simple source-combine is also less competitive in domain adaptation because this ideology does not consider the domain shift exists between multiple source domains.

Table 2: Comparison of classification accuracy (%) on Office-31 dataset. The red and blue texts denote the optimal and suboptimal results.

Standards	Methods	A, W → D	A, D → W	D, W → A	Avg
Single Best	ResNet	99.3	96.7	62.5	86.2
	DDC	98.2	95.0	67.4	86.9
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGrad	99.1	96.9	68.2	88.1
	RTN	99.4	96.8	66.2	87.5
	(CVPR'19) GCAN	99.8	97.1	64.9	87.3
	(CVPR'21) FixBi	100.0	99.3	79.4	92.9
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGrad	99.7	98.1	67.6	88.5
Multi Source	(CVPR'18) DCTN	99.3	98.2	64.2	87.2
	(AAAI'19) MFSAN	99.5	98.5	72.7	90.2
	(IJCNN'20) MDADFRE	99.6	98.7	73.1	90.5
	(arXiv'21) SImpAl	99.2	97.4	70.6	89.0
	(TNNLS'21) MSCLDA	99.8	98.8	73.7	90.8
	Ours	99.6	99.3	86.1	95.0

5) Network complexity analysis. The number of network parameters in this paper is counted excluding the pretrained parameters. Apart from the methods listed above, DCTN is testified consuming 119.02M parameters and M-DAN which is lacking implementation details isn't engaged. As shown in Fig. 4, network parameter on Office-Home of the proposed MDA-GM (3.7M) is smaller than CMSDA (4.85M) and MSCLDA (7.13M). Compared with baseline MFSAN (3.56M), the GCN feature space and multiple alignment strategies we engaged bring only a little higher computation but increase classification accuracy by 6% on Office-Home. It is also worth mentioning that DARN (2.61M), MDMN (2.61M), and M3SDA (2.62M) with similar model structure largely differs in their performance. This could be interpreted as the model optimization strategy is the main determinant in MDA. Similarly, Our method achieves the best performance with moderate network parameters on Office-31. Although our method is not the most light-weight method among all, good trade-off between computational complexity and classification performance is achieved.

Table 3: Comparison of classification accuracy (%) on Office-Home dataset. The red and blue texts denote the optimal and suboptimal results.

Standards	Methods	C, P, R → A	A, P, R → C	A, C, R → P	A, C, P → R	Avg
Single Best	ResNet	65.3	49.6	79.7	75.4	67.5
	DDC	64.1	50.8	78.2	75.0	67.0
	DAN	68.2	56.5	80.3	75.9	70.2
	D-CORAL	67.0	53.6	80.3	76.3	69.3
	RevGrad	67.9	55.9	80.4	75.8	70.0
	(CVPR'19) GCAN	50.1	49.1	72.5	64.5	59.1
	(CVPR'21) FixBi	76.4	62.9	86.7	81.7	76.9
Source Combine	DAN	68.5	59.4	79.0	82.5	72.4
	D-CORAL	68.1	58.6	79.5	82.7	72.2
	RevGrad	68.4	59.1	79.5	82.7	72.4
Multi Source	(NIPS'18) MDAN	68.1	67.0	81.0	82.8	74.7
	(NIPS'18) MDMN	68.7	67.8	81.4	83.3	75.3
	(ICCV'19) M^{3}SDA	64.1	62.8	76.2	78.6	70.4
	(AAAI'19) MFSAN	72.1	62.0	80.3	81.8	74.1
	(ICML'20) DARN	70.0	68.4	82.7	83.9	76.2
	(arXiv'21) SImpAl	70.8	56.3	80.2	81.5	72.2
	(TNNLS'21) MSCLDA	71.6	61.4	79.9	80.6	73.4
	(arXiv'21) CMSDA	71.5	67.7	84.2	83.0	76.6
	Ours	77.0	68.9	88.6	86.1	80.1

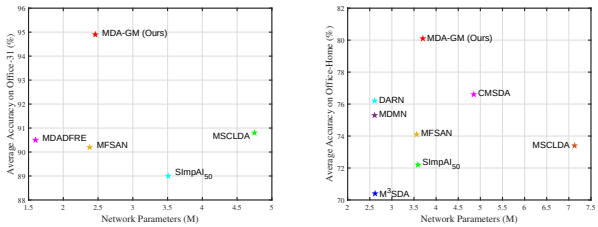
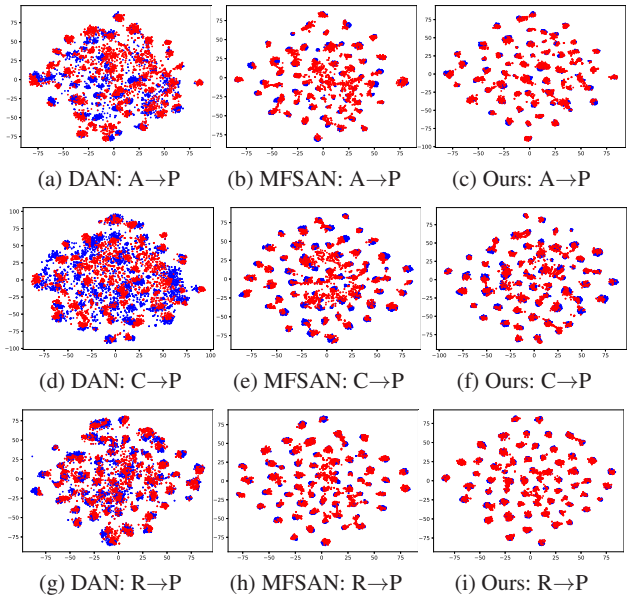


Fig. 4: Comparison of network performance and parameters on Office-31 (left) and Office-Home (right) datasets.

6) Feature visualization. We visualize the latent feature space representation of source and target domains as well as their categories with t-SNE [23]. Since the final results are obtained from each source domain classifier, we can visualize them individually. Fig. 5 shows the feature alignment performance on Office-Home. It is obvious that domain alignment results of MFSAN and MDA-GM are much better than that of DAN, which is consistent with the quantitative analysis in Table. 3. The matching of source features and target features of MDA-GM is more accurate and has the most obvious clear decision boundary. Fig. 6 show the t-SNE visualization of classification objects in target domain. MDA-GM has higher category discrimination than MFSAN and DAN, which indicates that it has better capability to adapt to datasets with large shift.

6) Ablation analysis. We reserved the domain alignment and classification result alignment provided by existing studies and compared the adaptation performance of

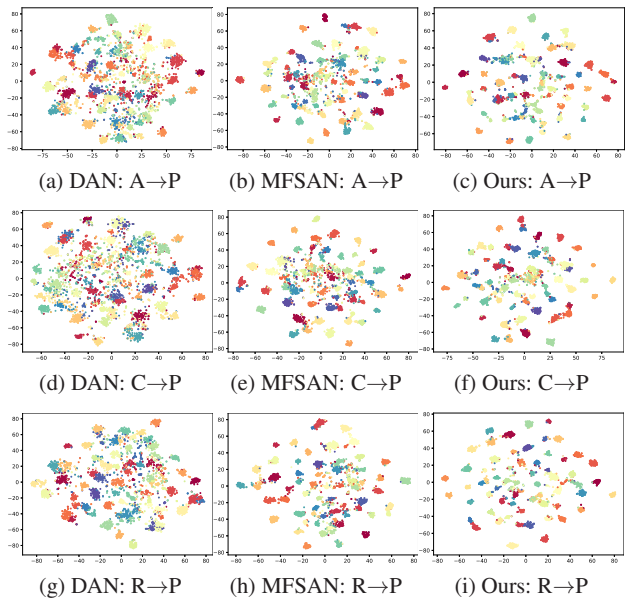


(a) DAN: A→P (b) MFSAN: A→P (c) Ours: A→P

(d) DAN: C→P (e) MFSAN: C→P (f) Ours: C→P

(g) DAN: R→P (h) MFSAN: R→P (i) Ours: R→P

Fig. 5: t-SNE visualization of latent space representation on Office-Home. The blue dot represents the source feature and the red dot represents the target feature.



(a) DAN: A→P (b) MFSAN: A→P (c) Ours: A→P

(d) DAN: C→P (e) MFSAN: C→P (f) Ours: C→P

(g) DAN: R→P (h) MFSAN: R→P (i) Ours: R→P

Fig. 6: t-SNE visualization of target domain on Office-Home. Categories are indicated in different colors.

MDA-GM with (w/) or without (w/o) category alignment and structure alignment. The results are shown in Fig. 7. It is obvious that category alignment and structure alignment largely promote MDA, and structure alignment has a better performance improvement than category alignment. It could be seen that all variants seem to achieve similar performance on A, W→D and A, D→W while ablating each of the alignment presents a noticeable decrease on D, W→A performance. This verifies that introducing GCN-based latent space mapping with a corresponding structure alignment largely benefits adaptation tasks. As the category alignment is set to optimize category decision boundary, ablating category alignment results in the least decrease in accuracy (1.7% on Office-31 and 3.2% on Office-Home). Abandoning structure alignment would lower classification accuracy more noticeably as structure information reflects the inherent properties of data. This alignment ensures data with the same structure in latent feature space can be more distinguished from data with different structures. This verifies the effectiveness of adopting GCN feature extraction, designing structure alignment and additional category alignment in a unified MDA network.

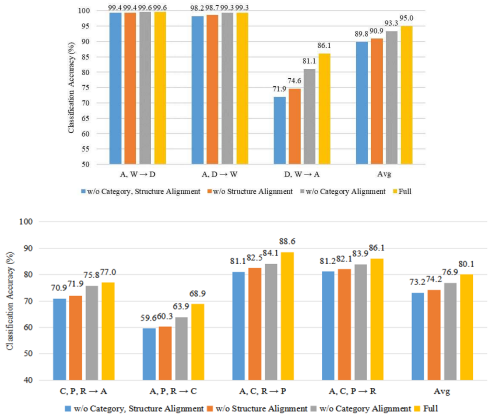


Fig. 7: Ablation results of MDA-GM on Office 31 (up) and Office-Home (down) datasets.

4 Conclusions

This paper proposes an MDA-GM algorithm that uses domain labels, category labels and data structure information for image classification. GCN-based latent feature space is engaged in MDA together with conventional CNN-based latent feature space, which extracts domain data feature with rich structure information. Apart from existing domain alignment and classification result alignment, we introduce two novel alignment strategies generally applicable to structural level and category level features. In detail, the structure alignment ensures data with the same structure in latent feature space closely clustered. This alignment strategy can specifically improve classification performance on adaptation tasks with large domain shift. The category alignment further enhances category-discriminative ability and optimizes category decision boundary. Comparative experiments on two standard datasets show that the proposed

method brings sufficient improvement among state-of-the-art MDA methods.

References

1. Noori, S., Tahmoresnezhad, J.: Joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation. Signal Image Video Process. **15**(2), 279-287 (2021)
2. Xu, R., Chen, Z., Zuo, W., et al.: Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3964-3973 (2018)
3. Zhu, Y., Zhuang, F., Wang, D.: Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In Proc. AAAI Conf. Artif. Intell., pp. 5989-5996 (2019)
4. Zhao, S., Li, B., Xu, P., et al.: MADAN: multi-source adversarial domain aggregation network for domain adaptation. Int. J. Comput. Vis., **129**, 2399-2424 (2021)
5. Ma, X., Zhang, T., Xu, C.: GCAN: Graph convolutional adversarial network for unsupervised domain adaptation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 8266-8276 (2019)
6. Peng, X., Bai, Q., Xia, X., et al.: Moment matching for multi-source domain adaptation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1406-1415 (2019)
7. Li, K., Lu, J., Zuo, H., et al.: Multi-source contribution learning for domain adaptation. IEEE Trans. Neural Netw. Learn. Syst. **99**, 1-15 (2021)
8. Tzeng, E., Hoffman, J., Zhang, N., et al.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
9. Long, M., Cao, Y., Wang, J., et al.: Learning transferable features with deep adaptation networks. In Int. Conf. Mach. Learn., pp. 97-105 (2015)
10. Saenko, K., Kulis, B., Fritz, M., et al.: Adapting visual category models to new domains. In Proc. European Conf. Comput. Vision., pp. 213-226 (2010)
11. Venkateswara, H., Eusebio, J., Chakraborty, S., et al.: Deep hashing network for unsupervised domain adaptation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 5018-5027 (2017)
12. Venkat, N., Kundu, J. N., Singh, D., et al.: Your classifier can secretly suffice multi-source domain adaptation. arXiv preprint arXiv:2103.11169 (2021)
13. Li, K., Lu, J., Zuo, H., et al.: Multi-source domain adaptation with distribution fusion and relationship extraction. In Int. Joint Conf. on Neural Networks, pp. 1-6 (2020)
14. Zhao, H., Zhang, S., Wu, G., et al.: Adversarial multiple source domain adaptation. In Proc. Int. Conf. on Neural Inform. Process., pp. 8559-8570 (2018)
15. Li, Y., Carlson, D. E.: Extracting relationships by multi-domain matching. In Proc. Int. Conf. on Neural Inform. Process., pp. 6799-6810 (2018)
16. Wen, J., Greiner, R., Schuurmans, D.: Domain aggregation networks for multi-source domain adaptation. In Int. Conf. Mach. Learn., pp. 10214-10224 (2020)
17. Scalbert, M., Vakalopoulou, M., Couzini-Devy, F., et al.: Multi-source domain adaptation via supervised contrastive learning and confident consistency regularization. arXiv preprint arXiv:2106.16093 (2021)
18. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 770-778 (2016)
19. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In Proc. European Conf. Comput. Vision. pp. 443-450 (2016)
20. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation, In Proc. Int. Conf. Mach. Learn., pp. 1180-1189 (2015)
21. Long, M., Zhu, H., Wang, J., et al.: Unsupervised domain adaptation with residual transfer networks. arXiv preprint arXiv:1602.04433 (2016)
22. Na, J., Jung, H., Chang, H. J., et al.: FixBi: Bridging domain spaces for unsupervised domain adaptation. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1094-1103 (2021)
23. Van, der, Maaten. L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(2605), 2579-2605.(2008)