Pattern Recognition 140 (2023) 109533

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Reciprocal normalization for domain adaptation

Zhiyong Huang[a], Kekai Sheng[b], Ke Li[b], Jian Liang[c], Taiping Yao[b], Weiming Dong[c], Dengwen Zhou[a], Xing Sun[b,*]

[a] School of Control and Computer Engineering, North China Electric Power University, China
[b] Youtu lab, Tencent, Shanghai, China
[c] NLPR, Institute of Automation, Chinese Academy of Sciences and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 7 December 2021
Revised 28 September 2022
Accepted 12 March 2023
Available online 14 March 2023

Keywords:
Domain adaptation
Feature normalization
Deep neural network

ABSTRACT

Batch normalization (BN) is widely used in modern deep neural networks, which has been shown to represent the domain-related knowledge, and thus is ineffective for cross-domain tasks like unsupervised domain adaptation (UDA). Existing BN variant methods aggregate source and target domain knowledge in the same channel in normalization module. However, the misalignment between the features of corresponding channels across domains often leads to a sub-optimal transferability. In this paper, we exploit the cross-domain relation and propose a novel normalization method, Reciprocal Normalization (RN). Specifically, RN first presents a Reciprocal Compensation (RC) module to acquire the compensatory for each channel in both domains based on the cross-domain channel-wise correlation. Then RN develops a Reciprocal Aggregation (RA) module to adaptively aggregate the feature with its cross-domain compensatory components. As an alternative to BN, RN is more suitable for UDA problems and can be easily integrated into popular domain adaptation methods. Experiments show that the proposed RN outperforms existing normalization counterparts by a large margin and helps *state-of-the-art* adaptation approaches achieve better results. The source code is available on https://github.com/Openning07/reciprocal-normalization-for-DA.

1. Introduction

Unsupervised domain adaptation (UDA) [1–5] aims to transfer the knowledge learned from the labeled source domain to the unlabeled target domain. It has been widely applied in classification [6], detection [7], and segmentation [8]. Technically, besides prevailing feature alignment [1,9,10] and pixel-level image translation [11,12], to enhance the feature transferability and learn domain-specific knowledge better, many researchers (*e.g.*, [13–16]) focus on improving the feature normalization module in deep neural networks (DNNs) to narrow the domain gap.

Technically, batch normalization (BN) [17] is a powerful approach to alleviate the internal covariate shift and has been widely used in DNNs, *e.g.*, ResNet-50 [18]. Nevertheless, recent research works [15,16] point out that BN suffers from losing domain-specific information in the UDA scenario, because sharing the mean and variance for the two domains are inappropriate [15]. To compensate for the deficiency of BN, several methods are proposed to preserve the domain-specific knowledge [13–16]. AdaBN [13] uses different domain statistics for the two domains. However, only employing the target statistics in the inference can lose the information of the source domain. To merge the information of different domains, AutoDIAL [14] fuses domain statistics channel by channel using a shared weight parameter for each channel. TN [15] proposes a channel attention mechanism to highlight the channels with high transferability to further focus on the important information.

The aforementioned methods reinforce UDA by aggregating the domain knowledge extracted from the corresponding channels. For different examples from the same domain, the learned patterns are likely to be captured by the same channel (see the upper and middle rows in Fig. 1). When encountering cross-domain scenarios, we observe that the same or similar patterns cannot always be captured by the same channel, however, which is always ignored by existing UDA methods. As illustrated in Fig. 1 (c) and (f), different patterns are captured by the same channels across domains. Thus, merging the domain knowledge of corresponding channels across domains in [14] can inevitably lose domain-specific information and lead to sub-optimal UDA performance. Another important observation is that similar patterns from different domains are likely to exist in the non-corresponding channels (*e.g.*, Fig. 1(b) and (e)).

* Corresponding author.
E-mail address: support@elsevier.com (X. Sun).
URL: http://www.elsevier.com (Z. Huang)

https://doi.org/10.1016/j.patcog.2023.109533
0031-3203/

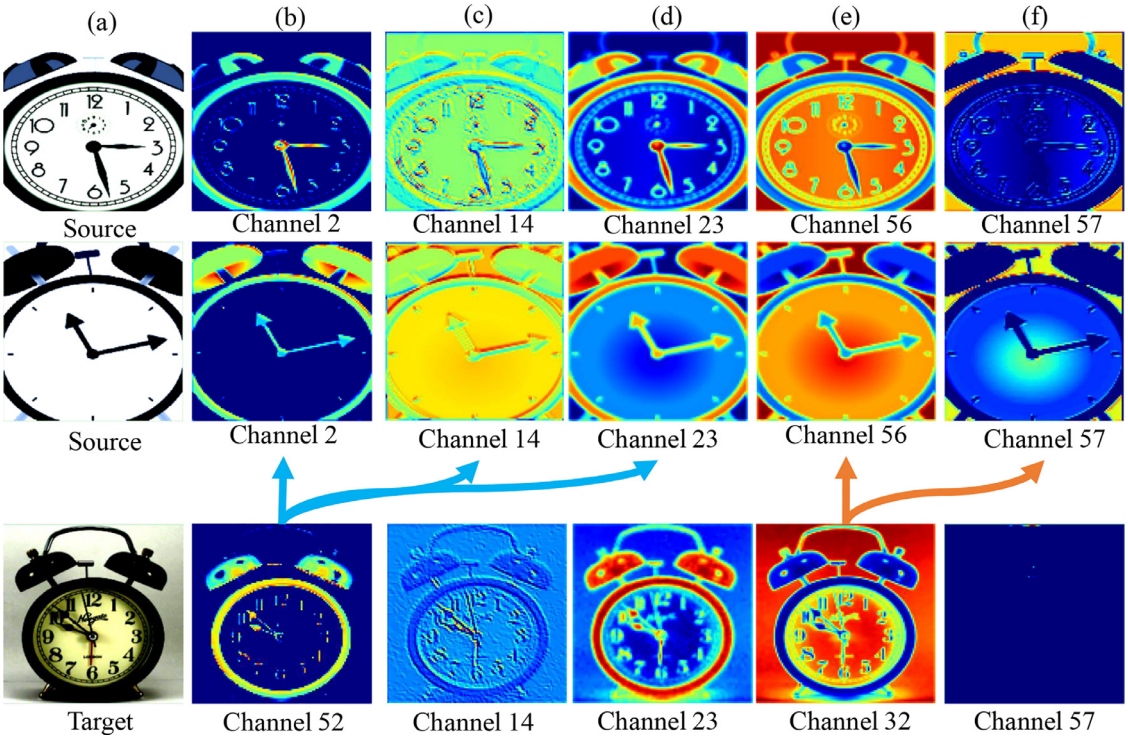


Fig. 1. The visualization of feature maps from the first ReLU layer of ResNet-50 [18] on the UDA task Clipart (1st and 2nd rows) → Art (3rd row), which is trained with CDAN [9]. (a) Three images of Alarm Clock. (d) is the similar pattern at the same channel. (c) and (f) are different patterns at the same channels. (b) and (e) are the similar patterns at the different channels.

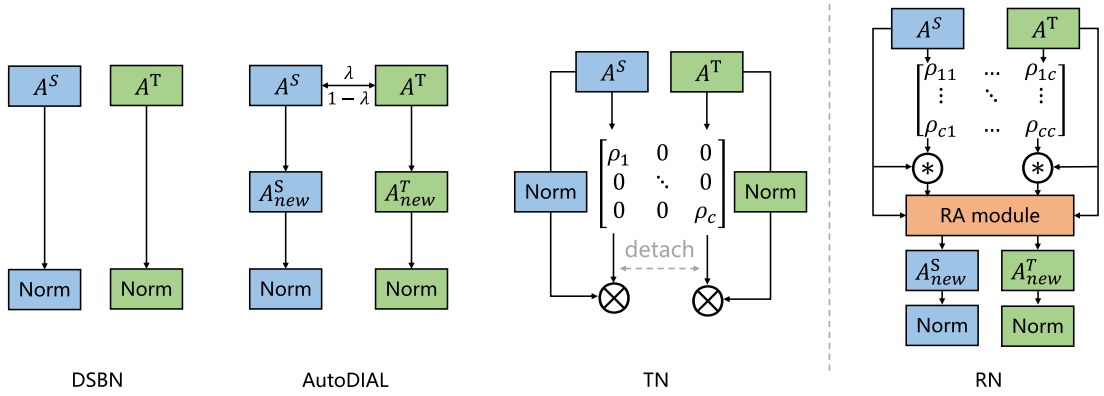


Fig. 2. The differences between our RN and other typical UDA normalization technologies. The A^S and A^T denote the data statistics of source and target domains, respectively. Specifically, DSBN [16] adopts totally separated normalization modules for each domain. AutoDIAL [14] and TN [15] consider the correlations of corresponding cross-domain channels to enhance the transferability. Our RN captures the long-range correlations between cross-domain channels. The probability of TN is detached from calculation graph. Our RN utilizes a Reciprocal Aggregation (RA) module to adaptively aggregate both source and target information.

Moreover, the shareable patterns at non-corresponding channels across domains are not just the one-to-one relationship, as shown in Fig. 1 (the orange and blue arrows). Therefore, adaptively considering the correlations of all cross-domain channels is crucial to break through the bottleneck in DA architectures.

Building on the observations and deductions above, in this paper, we propose a novel Reciprocal Normalization (RN) scheme for unsupervised domain adaptation. Figure 2 illustrates the key differences between existing UDA normalization techniques and our RN. In contrast to the local behavior of BN and its variants towards domain adaptation, the proposed RN is able to capture long-range correlations directly by computing interactions between any two cross-domain channels and then conducts reciprocity between domains during normalization. Specifically, we firstly present a reciprocal compensation (RC) module to acquire the compensatory of each source/target channel for the counterpart in the target/source domain by modeling the correlation of any two cross-domain channels. For efficient reciprocity and effective domain alignment, we then develop a Reciprocal Aggregation (RA) module to adaptively aggregate the feature with its cross-domain compensatory component. Put RC and RA together, we propose RN to boost the performance of various domain adaptation tasks.

In summary, our main contributions are three-fold:

- We propose a novel RN scheme for domain adaptation to address the issue of channel misalignment across domains and get better result on the target domain.
- The proposed RN structurally aligns the source and target domains by conducting reciprocity across domains. Besides being

a plug-and-play module, RN can be also integrated with other domain adaptation methods to achieve better results.
- Experiments on three benchmarks (ImageCLEF-DA, Office-Home, and VisDA-C) and various DA scenarios (closed-set DA, partial-set DA, and multi-source DA) indicate that our RN outperforms existing normalization methods and benefit domain adaptation approaches in various scenarios.

2. Related work

2.1. Domain adaptation

Existing approaches mainly focus on loss function design or network design. Technically, the loss function design usually starts from two directions. i) To match all statistics of the two domains to minimize cross-domain distribution discrepancy: DDC [19] and DAN [20] employ Maximum Mean Discrepancy (MMD) [21] to measure and reduce the discrepancy of source and target domains; JAN [6] utilizes Joint Maximum Mean Discrepancy to combine adversarial learning with MMD; SWD [22] introduces Sliced Wasserstein Distance and CAN [2] leverages Contrastive Domain Discrepancy to find a better measure of the domain discrepancy. ii) To introduce domain discriminators and exploit adversarial learning to encourage domain confusion: DANN [1] introduces domain adversarial loss to learn domain-invariant representations; ADDA [23] combines adversarial learning with discriminative feature learning via adopting asymmetric feature extractors for each domain; CDAN [9] employs a conditional domain-adversarial paradigm to train an adversarial adaptation model. More recently, advanced loss functions (*e.g.*, BSP [24], IAA [25], BNM [26], and SRDC [27]), learning schemes [28–32] and new network designs (*e.g.*, TN [15], DCAN [33], and BCDM [34]) are proposed for better performance on target domain.

However, all these existing methods overlook the misalignment between the features of corresponding channels across domains, which often leads to a sub-optimal DA performance. Additionally, as a general method, our work is able to benefit many unsupervised domain adaptation scenarios including vanilla closed-set UDA, partial-set DA (PDA), and multi-source DA (MSDA).

2.2. Normalization techniques

It is widely applied in CNNs to make them learn faster, more stable, and increase their generalization ability [15,35,36]. Representative methods include BN [17], LN [37], AdaBN [13], GN [38], SN [39], EvoNorm [40], and Representative Normalization [41]. For better domain adaptation, researchers have devised novel designs to mitigate the shortcomings in BN. AdaBN [13] uses the statistics of source domain during training and those of target domain during evaluation, respectively. AutoDIAL [14] integrates the statistics of two domains channel by channel in order to align the source and target feature distributions. Domain Specific BN (DSBN) [16] normalizes the source and target representations completely individually, including affine parameters. Transferable Normalization (TN) [15] utilizes the statistics of two domains to calculate corresponding channel attention, which are all detached from the computation graph. ConvNorm [42] proposes an adaptation layer $\mathcal{A}$ to whiten and color source domain data, then $\mathcal{A}$ is fine-tuned on the target domain. DWT [43] uses two covariance matrices to whiten feature maps from source and target domains, respectively. Particularly, DSBN [16] and DWT [43] adopt totally separately normalize feature maps from source and target domains. AutoDIAL [14] and TN [15] consider the corresponding cross-domain channels to enhance the transferability. They achieve promising progresses but neglect the misalignment between non-corresponding channels across domains.

Different from these existing counterparts, we focus on modeling the non-corresponding channels in CNNs for domain adaptation. We propose a novel feature normalization method to facilitate domain alignment via conducting cross-domain reciprocity.

3. Methodology

In this section, we present the details of our RN. We firstly revisit the BN and reformulate it for clear presentation (Section 3.1). Then, we introduce RN to alleviate the misalignment between features across domains (Section 3.2).

3.1. Revisiting batch normalization

BN [17] is excellent in CNNs for many visual recognition tasks. Technically, the BN layer firstly estimates the standardized features (*i.e.* with zero mean and unit standard deviation) at the channel dimension on the basis of mini-batch data, and then scales and shifts the standardized features by using a pair of learnable parameters γ and β. Given the feature $x \in \mathbb{R}^{N\times C\times H\times W}$, the transformed $\widehat{x}$ is acquired through BN layer as:

$$\widehat{x}^{(i)} = \gamma\hat{x} + \beta, \quad \hat{x} = \frac{x-\mu}{\sqrt{(\sigma^2)+\epsilon}}, \tag{1}$$

where ϵ is a small constant for numerical stability. μ and σ^2 are the *mean* and *variance* statistics for each channel over a mini-batch respectively, and are defined as:

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu^2). \tag{2}$$

To obtain the accumulated statistics for the whole training data, the BN layer keeps running estimates towards the μ and σ^2 to obtain $\bar{\mu}$ and $\bar{\sigma}^2$ during training phase:

$$\bar{\mu}^{t+1} = (1-\alpha)\bar{\mu}^t + \alpha\mu^t, \quad (\bar{\sigma}^{t+1})^2 = (1-\alpha)(\bar{\sigma}^t)^2 + \alpha(\sigma^t)^2, \tag{3}$$

where α denotes the momentum and t is the index of mini-batch data. The estimated mean and variance will be used to normalize the features during inference phase. In this way, the BN layer can successfully accelerate and stabilize training. However, it is somewhat unreasonable to directly share the same mean and variance statistics between source and target domains since there exists a significant gap.

3.2. Reciprocity normalization (RN)

Several methods have recently been proposed to address the limitation of BN, such as AdaBN [13], AutoDIAL [14], DWT [43], DSBN [16], and TN [15]. We illustrate the main differences between other typical UDA normalization techniques and our RN in Fig. 2. Generally, those methods all adopt separate normalization to avoid sharing exactly the same mean and variance. However, such a mechanism suffers from another problem, *i.e.*, the misalignment of activations in the corresponding channel across domains, which sometimes leads to negative transfer. Due to the differences in background, style, distribution, *e.t.c.*, between domains, it is intuitive that similar patterns of source and target domains are likely to be activated by non-corresponding cross-domain channels (*e.g.*, Fig. 1(c) and (f)). As a result, simply normalizing source and target features separately may lose the domain information. Although AutoDIAL and TN care for the information of corresponding channels, they only partially alleviate the problem since they neglect the correlation between non-corresponding channels.

Motivated by the aforementioned observations and analyses, we propose a novel RN method for domain adaptation. The main

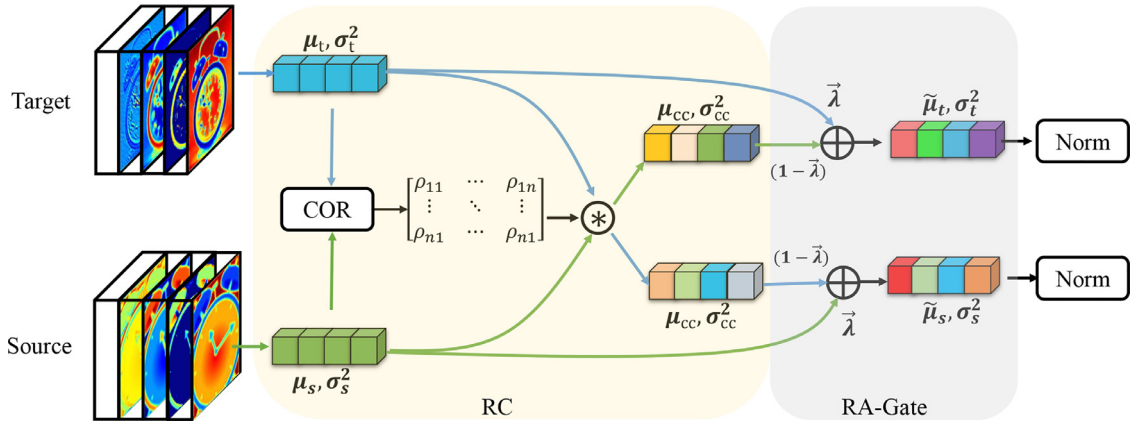


Fig. 3. A main schematic diagram of RN. The blue → and the green → denote the target and the source information flows, respectively. The "*COR*" denotes calculating the correlations between cross-domain channels. $\vec{\lambda}$ denotes the weight vector of reciprocal aggregation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

pipeline of RN is shown in Fig. 3. It consists of two main procedures: RC and RA. For a convenient and concise expression, we only present the reciprocity from the source domain to the target domain, and the other half of the corresponding operation is basically the same.

3.2.1. Reciprocal compensation (RC)

It models the relationship of any two channels across domains. The key insight is that similar patterns between domains are likely to be captured by not only the corresponding but also non-corresponding channels across domains (*i.e.*, one-to-more relationship) when the domain shift is significant. We aim to fully consider any two cross-domain channels and then conduct reciprocity between domains.

Specifically, we first calculate the source (s) and target (t) statistics μ_s, σ_s^2 and μ_t, σ_t^2 via Eq. (2). To enable the channels with similar characteristics to have more correlation, we compute the correlation between any two channels via the negative l_2 distance:

$$E^{\mu}_{i,j} = -(\mu_{i,t} - \mu_{j,s})^2, \quad E^{\sigma^2}_{i,j} = -(\sigma^2_{i,t} - \sigma^2_{j,s})^2, \tag{4}$$

where $E_{i,j}$ denotes the correlation between the ith channel of target domain and the jth channel of source domain. Below we use $E^{\mu}_{t\to s}$ and $E^{\sigma^2}_{t\to s}$ to denote the two correlation matrices. In Section 4.7, we compare the results of some popular distance measures and find that l_2 distance performs the best, thus we choose l_2 distance as our default setting.

Then, to obtain the probabilistic weights of correlation between any two cross-domain channels, we normalize $E^{\mu}_{t\to s}$ and $E^{\sigma^2}_{t\to s}$ at the row dimension with softmax layer. The correlation score matrices $\rho^{\mu}_{t\to s}$ and $\rho^{\sigma^2}_{t\to s}$ can be computed respectively via:

$$\rho^{\mu}_{t\to s} = softmax(E^{\mu}_{t\to s}, \texttt{dim=1}), \quad \rho^{\sigma^2}_{t\to s} = softmax(E^{\sigma^2}_{t\to s}, \texttt{dim=1}), \tag{5}$$

where `dim=1` denotes the normalization of the matrices at the row dimension. In this way, we obtain the normalized correlation probability between each channel of target domain and all channels of source domain. This appears similar to TN [15] that quantifies the transferability of corresponding channels across domains to calculate the channel attention weights. However, TN neglects the correlation between non-corresponding channels across domains. Usually, the limitation of misalignment of channels can be partially mitigated by TN, but TN just puts a large emphasis on the corresponding channels with similar patterns and neglects to fully exploit the similar patterns in non-corresponding channels. It also leads to the loss of the domain information of corresponding channels with different patterns to a certain extent.

Finally, the compensatory of each channel of target domains can be computed in the source domain space. The compensatory of μ_t and σ_t^2 can be obtained by:

$$\mu_{t,cc} = \rho^{\mu}_{t\to s} \cdot \mu_s, \quad \sigma^2_{t,cc} = \rho^{\sigma^2}_{t\to s} \cdot \sigma^2_s. \tag{6}$$

Such calculation allows each compensatory of channels to capture long-range correlations directly by conducting reciprocity among all cross-domain channels, including similar and complementary channels.

Furthermore, the global domain information is exploited by RC beyond the limit of the local receptive field of the convolutional kernel.

3.2.2. Reciprocal aggregation (RA)

Although we have obtained the compensatory for each channel of the target domain in the source domain space, it is inappropriate to directly utilize μ_{cand} and σ^2_{cand} to conduct the feature normalization since it may cause the loss of original domain-specific knowledge. The empirical results in Section 4.7 also verify this judgment. Thus, we aim to enable our module to adaptively learn the degree of reciprocity of domain information from various deep layers. Particularly, AutoDIAL [14] directly integrates the statistics of two domains via one single 1-D parameter to endow the network with the ability to automatically align source and target domains. We follow this strategy and develop RA to adaptively aggregate the matched compensatory and the original domain statistics. Specifically, we introduce the learnable gate parameters $g \in [0.5, 1]^C$:

$$\widetilde{\mu}_t = g^{\mu}_t * \mu_t + (1 - g^{\mu}_t) * \mu_{t,cc}, \quad \widetilde{\sigma}^2_s = g^{\sigma^2}_t * \sigma^2_t + (1 - g^{\sigma^2}_t) * \sigma^2_{t,cc}, \tag{7}$$

where "$*$" denotes the Hadamard product. During training, g is initialized as a unit vector so that RN performs the pure domain-specific normalization at the beginning of training, and then reduces the domain discrepancy via bridging the gaps between the source and target domains with g updated progressively. Due to such an aggregation between each channel and its compensatory, the mutual domain information associated with individual channels can be emphasized accordingly. Different from AutoDIAL [14] that directly mixes statistics of the two domains, RN uses RC to produce the information fed into aggregation It considers the correlation between any two cross-domain channels so as

to contain more domain information. Besides, AutoDIAL uses a single 1-D parameter to align all the domain statistics, which may be less effective and adaptive. By contrast, the mean and variance are equipped with their own C-D parameters, endowing RN with the ability to adaptively learn where and how to conduct aggregation.

Algorithm 1: The forward pass of RN during training.

Require: Feature maps of source and target domains in a mini-batch: $\{x_s, x_t\} \in \mathbb{R}^{N\times C\times H\times W}$; learnable g in RA: $\{g_s^{\mu}, g_s^{\sigma^2}, g_t^{\mu}, g_t^{\sigma^2}\} \in [0.5, 1]^C$; learnable affine parameters: γ. β.
Ensure: $\widehat{x}_s$, $\widehat{x}_t$
Calculate $\{\mu_s, \mu_t, \sigma_s^2, \sigma_t^2\} \in \mathbb{R}^{C\times 1}$
For concise expression, let $z \in \{\mu, \sigma^2\}$
Calculate the correlation strength:
$E_{t\to s}^z \leftarrow -(z_{i,t} - z_{j,s})^2$,
$E_{s\to t}^z \leftarrow (E_{t\to s}^z)^T$
Obtain the probabilistic weights of correlation:
$\rho_{t\to s}^z \leftarrow softmax(E_{t\to s}^z, \texttt{dim=1})$
$\rho_{s\to t}^z \leftarrow softmax(E_{s\to t}^z, \texttt{dim=1})$
Obtain the candidates of each statistics:
$z_{t,cc} \leftarrow \rho_{t\to s}^z \cdot z_s$
$z_{s,cc} \leftarrow \rho_{s\to t}^z \cdot z_t$
Reciprocal aggregation:
$\tilde{z}_t \leftarrow g_t^z * z_t + (1 - g_t^z) * z_{t,cc}$
$\tilde{z}_s \leftarrow g_s^z * z_s + (1 - g_s^z) * z_{s,cc}$
$\widehat{x}_s \leftarrow \gamma \frac{x_s - \tilde{\mu}_s}{\sqrt{\tilde{\sigma}_s^2 + \epsilon}} + \beta$, $\widehat{x}_t \leftarrow \gamma \frac{x_t - \tilde{\mu}_t}{\sqrt{\tilde{\sigma}_t^2 + \epsilon}} + \beta$

3.2.3. Separate normalization

Without loss of generality, we adopt the aggregated domain statistics to normalize the feature representations from source and target domains, separately. Akin to BN, we utilize affine parameters γ and β to re-scale and re-shift the normalized feature responses, where γ and β are shared in the two domains. Here, we just present the normalization of target domain as follows:

$$\widehat{x}_t^{(i)} = \gamma^{(i)}\hat{x}_t^{(i)} + \beta^{(i)}, \quad \hat{x}^{(i)} = \frac{x_t^{(i)} - \widetilde{\mu}^{(i)}}{\sqrt{\widetilde{\sigma}_t^{2(i)} + \epsilon}}, \tag{8}$$

where ϵ is a small constant to avoid divide-by-zero. In this way, the domain-specific information can be well captured at the early training stage and the alignment between the two domains can be progressively carried out via adaptive reciprocity.

3.2.4. Inference

To reduce time cost at inference, we adopt a memory strategy similar to BN. During training, RN keeps running estimates of its aggregated mean and variance of each domain, via exponential moving average with a hyper-parameter α, which is given by:

$$\bar{\mu}_d^{t+1} = (1-\alpha)\bar{\mu}_d^t + \alpha\widetilde{\mu}_d^t, \quad (\bar{\sigma}_d^{t+1})^2 = (1-\alpha)\bar{\sigma}_d^t + \alpha(\widetilde{\sigma}_d^t)^2, \tag{9}$$

where $d \in \{s, t\}$, α is initialized to 0.1, and the estimated aggregated statistics $\bar{\mu}_t$ and $\bar{\sigma}_t^2$ are used for the examples from target domain at inference. Such a strategy allows RN directly utilize the estimated domain statistics to normalize the examples during the evaluation without performing secondary calculations about RC and RA.

4. Experiments

In this section, we evaluate the proposed RN on three benchmarks of three adaptation scenarios: vanilla closed-set UDA, partial-set DA (PDA), and multi-source DA (MSDA). We compare the performance of RN and the other existing normalization approaches. Besides, we conduct ablation study of the two modules (*i.e.*, RC and RA) in our RN. To better understand the rationale and the working mechanism of RN, we have some theoretical analyses based on quantitative results and feature visualization.

4.1. Setup

Datasets We experiment on three cross-domain benchmarks. i) *ImageCLEF-DA* is a small-scale dataset with 12 classes shared by 3 domains: Caltech-256 (C), ILSVRC 2012 (I), and Pascal VOC 2012 (P). We conduct experiments on all the 6 transfer tasks. ii) *Office-Home* [44] is a medium-sized benchmark of 12 adaptation tasks from 4 domains: Artistic (Ar), Clip Art (Cl), Product (Pr), and Real-World (Rw). Each domain contains 65 everyday object categories. iii) *VisDA-C* [45] is a challenging large-scale benchmark of 12-class synthesis-to-real adaptation task. The source domain contains 152*K* synthetic 3D images, and the target domain has 55*K* real object images.

Baselines Besides the existing normalization modules for domain adaptation (*i.e.*, BN [17], AutoDIAL [14], DSBN [16], and TN [15]), we also select popular *state-of-the-art* approaches as the baselines in three typical scenarios: i) On the vanilla closed-set *UDA*, we compare with DAN [20], DANN [1], JAN [6], MCD [46], CDAN [9], iCAN [47], DTA [48], DWT [43], BSP [24], AFN [49], CRST [50], CADA [51], MDD [52], CAN [2], BNM [26], DCAN [33], IAA [25], STAFF [53], GVB on CDAN (CDAN-GD) [54], DMRL [55], DANCE [56], DWL [31], and PAS [32]. ii) On the *PDA*, we compare with DANN [1], IWAN [57], SAN [58], AFN [49], ETN [59], BA^{3}US [60], DANCE [56], and JUMBOT [61]. iii) On the *MSDA*, we compare with DANN [1], D-CORAL [62], CDAN [9], MetaMCD [63] and SImpAl [64]. For fair comparison, we run the proposed method three times with different random seeds and record the average results. For the clear comparison between the proposed RN and the existing normalization counterparts [14–17], please refer to Table 4.

Implementation Details Without loss of generality, we adopt four popular methods as the test-beds: DANN [1], CDAN [9], ETN [59], and BA^{3}US [60]. On one backbone network (*e.g.*, ResNet-50 [18]) pretrained on ImageNet, we replace all the BN [17] within different intermediate layers in the backbone with our RN without changing the original settings. We initialize the parameters of RA to unit vectors and constrain their weights to be in the range [0.5,1]. It should be pointed out that the substitution works without an additional pre-training procedure on ImageNet dataset, and it is flexible for practical usage. The flexible replacement indicates the versatility of our RN.

We implement the RN via PyTorch [65]. For fair comparison, the training configurations (*e.g.*, batch-size, learning rate, optimization algorithm) are all the same as the original baselines except the normalization module which are replaced by our RN. We conduct the experiments of RN with 3 random seeds and report the average accuracies.

4.2. Results on small-scale dataset

First, we conduct the comparison experiments on ImageCLEF-DA, one popular small-scale cross-domain benchmark. We adopt ResNet-50 as the backbone network and choose DANN and CDAN as the test-bed methods. As listed in Table 1, on the average performance of 6 adaptation scenarios, our RN helps DANN and CDAN promote their classification accuracies by 3.0% and 1.5%, respectively. The results demonstrate the effectiveness of our RN. For the

Table 1
Accuracy (%) on ImageCLEF-DA for ResNet-50 based UDA. The best results are in **bold**.

Method	I→P	P→I	I→C	C→I	C→P	P→C	AVG
Source only	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN [20]	74.5	82.2	92.8	86.3	69.2	89.8	82.5
DANN [1]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN [6]	76.8	88.0	94.7	89.5	74.2	91.7	85.8
iCAN [47]	79.5	89.7	94.7	89.9	78.5	92.0	87.4
PAS [32]	78.3	92.0	95.1	90.5	75.5	95.5	87.8
CAN [2]	77.2	90.3	96.0	90.9	78.0	**95.6**	88.0
DMRL [55]	77.3	90.7	97.4	91.8	76.0	94.8	88.0
CADA [51]	78.0	90.5	96.7	92.0	77.2	95.5	88.3
DCAN [33]	**80.5**	91.2	95.7	91.8	77.2	93.3	88.3
DANN [1]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
DANN+RN	78.1	90.1	96.3	91.7	78.0	94.0	88.0
CDAN [9]	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7
CDAN+RN	78.6	**92.7**	97.2	**92.8**	**79.1**	94.8	**89.2**

Table 2
Accuracy (%) on Office-Home benchmark for ResNet-50-based UDA and PDA methods. The best results are in **bold**.

Closed-set UDA	Ar→Cl	Ar → Pr	Ar → Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	AVG
Source only	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
JAN [6]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
PAS [32]	52.2	72.9	76.9	58.4	68.1	69.7	58.3	47.4	76.6	67.1	53.5	77.6	64.9
DWT [43]	50.3	72.1	77.0	59.2	69.3	70.2	58.3	48.1	77.3	69.3	53.6	82.0	65.6
BSP [24]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
AFN [49]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
MDD [52]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
STAFF [53]	53.3	71.9	**80.2**	63.1	69.8	74.1	65.3	50.9	77.8	73.1	56.6	82.4	68.2
CDAN-GD [54]	55.3	74.1	78.2	62.4	72.6	71.8	63.8	54.1	80.1	73.1	58.7	83.6	69.0
DANCE [56]	54.3	**75.9**	78.4	64.8	72.1	73.4	63.2	53.0	79.4	73.0	58.2	82.9	69.1
DANN [1]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
DANN+RN	47.3	63.1	74.4	57.1	64.7	68.4	55.2	47.8	75.9	68.9	53.5	79.3	63.0
CDAN [9]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
CDAN+RN	**55.6**	72.6	78.1	**65.7**	**74.7**	**74.6**	**66.2**	**57.1**	**82.0**	**75.2**	**60.5**	**84.6**	**70.6**
PDA	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	AVG
Source only	46.3	67.5	75.9	59.1	59.9	62.7	58.2	41.8	74.9	67.4	48.2	74.2	61.3
DANN [1]	35.5	48.2	51.6	35.2	35.4	41.4	34.8	31.7	46.2	47.5	34.7	49.0	40.9
IWAN [57]	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6
SAN [58]	44.4	68.7	74.6	67.5	65.0	77.8	59.8	44.7	80.1	72.2	50.2	78.7	65.3
DANCE [56]	53.6	73.2	84.9	70.8	67.3	82.6	70.0	50.9	84.8	77.0	55.9	81.8	71.1
AFN [49]	58.9	76.3	81.4	70.4	73.0	77.8	72.4	55.3	80.4	75.8	60.4	79.9	71.8
JUMBOT [61]	62.7	77.5	84.4	**76.0**	73.3	80.5	74.7	60.8	85.1	80.2	**66.5**	83.9	75.5
ETN [59]	52.9	78.2	83.2	70.2	69.4	77.6	69.5	50.8	81.0	76.3	54.5	82.0	70.5
ETN+RN	56.1	79.5	87.2	74.8	68.2	79.4	77.0	52.2	83.9	**82.2**	58.7	83.5	73.6
BA^{3}US [60]	60.6	**83.2**	**88.4**	71.8	72.8	**83.4**	75.5	61.6	86.5	79.3	62.8	86.1	76.0
BA^{3}US+RN	**63.5**	**83.2**	88.3	72.8	**73.4**	**83.4**	**77.2**	**62.6**	**87.7**	80.8	63.6	**87.0**	**77.0**

comparisons of RN and existing normalization modules [14–16] on ImageCLEF-DA, please refer to Table 4.

4.3. Results on medium-scale dataset

Next, we summarize the results of UDA and PDA experiments on Office-Home in Table 2. For fair comparison in PDA scenario, we follow the protocol of ETN [59] and BA^{3}US [60].[1] Experimental results show that the proposed RN promotes the performance of CDAN by 4.8% in UDA. In PDA, our RN also benefits ETN by 3.1% and BA^{3}US by 1.0%. It is also noteworthy that in PDA experiments on Office-Home, there is a large semantic difference between the two domains: source domain contains 65 classes while target domain contains only 25 classes. Despite the differences, our RN still helps ETN achieve better performance in average accuracy (from 70.5% to 73.6%).

Consequently, these numerical results ensure the versatility of the proposed RN in DA. It is convincing that RN can consistently helps boost performance of UDA methods. For the visualization of learned visual feature (*e.g.*, tSNE [66]), please refer to Section 4.10.

[1] https://github.com/tim-learn/BA3US.

4.4. Results on large-scale dataset

To further demonstrate the effectiveness of the proposed RN, we conduct evaluation experiments on VisDA-C benchmark. We compare several popular UDA methods and evaluate their classification accuracies in Table 3. We observe that our RN helps CDAN achieve better performance on ResNet-50 (by 9.6%) and ResNet-101 (by 6.2%). Additionally, the gap between CDAN+RN on ResNet-50 and that on ResNet-101 is only 0.5%, indicating our RN is particularly effective to conduct domain adaptation in the large-scale dataset. The possible reason is the domain statistics would be more accurate when the dataset is large, which is more beneficial to our RN.

4.5. Different normalization and general regularizer

To verify that our RN can work as a general regularizer in domain adaptation, we choose DANN [1] and CDAN [9] as the testbeds and conduct evaluation experiments on three benchmarks: Office-Home, ImageCLEF-DA, and VisDA-C. For fair comparisons, four *state-of-the-art* feature normalization modules [14–17] are considered. Noting that all following experiments of *Method+DSBN* are conducted without extra constraints, *e.g.*, pseudo labels algo-

Table 3
The average accuracies over 12 classes (%) on VisDA-C for closed-set UDA. The best results are in **bold**.

Method	ResNet-50	ResNet-101
Source only	-	52.4
JAN [6]	61.6	-
DAN [20]	61.6	62.8
MCD [46]	69.7	71.9
DMRL [55]	-	75.5
IAA [25]	75.8	-
BSP [24]	-	75.9
AFN [49]	-	76.1
DWL [43]	-	77.1
DANCE [56]	70.2	-
JUMBOT [61]	72.5	
CDAN-GD [54]	74.9	-
DTA [48]	76.2	-
DANN [1]	54.9	57.4
DANN+RN	71.4	74.9
CDAN [9]	70.0	73.9
CDAN+RN	**79.6**	**80.1**

rithm or other loss functions. The results are listed in Table 4, where "Y→X" means that the domain "Y" respectively adapts to other three domains of Office-Home and we report the average accuracy of the three transfer tasks.

As we observe in Table 4 that, our RN consistently offers larger improvements than other counterparts to a variety of domain adaptation methods on various datasets. It affirms the effectiveness of the proposed RN beyond existing normalization counterparts in DA. It is worth noting that DSBN [16] even produces worse performance on small- and medium-scale datasets, indicating that separating γ and β may suffers from *negative transfer* without the extra pseudo labels algorithms to conduct the target parameters update properly. In addition, when it comes to computation cost, our RN achieves better trade-off between the time costs of train and test phases simultaneously.

4.6. Results on MSDA

To demonstrate the versatility of our RN, we also conduct multi-source domain adaptation (MSDA) on Office-Home benchmark. We choose CDAN [9] as the baseline and also compare with existing normalization counterparts.

The results are listed in Table 5. For simplicity: "→X" denotes the adaptation task from other three domains to "X" domain. *Single Best* denotes the best performance of all tasks on single-source domain adaptation, and *Combination* refers to merging data from multiple source domains and constructing a new and larger source domain dataset. Obviously, our RN models consistently outperform the other BN-variant modules for all the settings. On the AVG, the proposed RN module promotes CDAN by 4.0% in *Single Best* scenario and 3.5% in *Combination* scenario. The improvements are higher than that from the existing normalization counterparts. CDAN+RN even outperforms several latest proposed methods for multi-source domain adaptation, such as D-CORAL [62], Meta-MCD [63], and SImpAl [64]. Consequently, the results indicate that RN is also versatile to benefit multi-source domain adaptation scenarios.

Table 5
Accuracy (%) on Office-Home for ResNet-50 based multi-source UDA. The best results are in **blod**.

Single Best	→Ar	→Cl	→Pr	→Rw	AVG
ResNet-50 [18]	53.9	41.2	59.9	60.4	53.9
D-CORAL [62]	67.0	53.6	80.3	76.3	69.3
RevGrad [1]	67.9	55.9	80.4	75.8	70.0
CDAN(+BN) [9]	70.9	56.7	81.6	77.3	71.6
CDAN+AutoDIAL [14]	71.2	57.5	81.4	76.2	71.6
CDAN+DSBN [16]	70.2	51.4	78.4	78.4	69.6
CDAN+TN [15]	71.9	59.0	82.9	79.5	73.3
CDAN+RN	**75.2**	**60.5**	**84.6**	**82.0**	**75.6**
Combination	→A	→Cl	→Pr	→Rw	AVG
ResNet-50 [18]	65.3	49.6	79.7	75.4	67.5
D-CORAL [62]	68.1	58.6	79.5	82.7	72.2
RevGrad [1]	68.4	59.1	79.5	82.7	72.4
Meta-MCD [63]	70.2	60.5	81.2	83.1	73.8
SImpAl [64]	73.4	62.4	81.0	82.7	74.8
CDAN(+BN) [9]	71.4	64.2	81.1	82.3	74.8
CDAN+AutoDIAL [14]	75.7	64.2	83.7	83.9	76.9
CDAN+DSBN [16]	71.7	57.2	77.5	79.1	71.4
CDAN+TN [15]	74.7	64.6	83.1	83.3	76.4
CDAN+RN	**75.6**	**66.8**	**85.3**	**85.3**	**78.3**

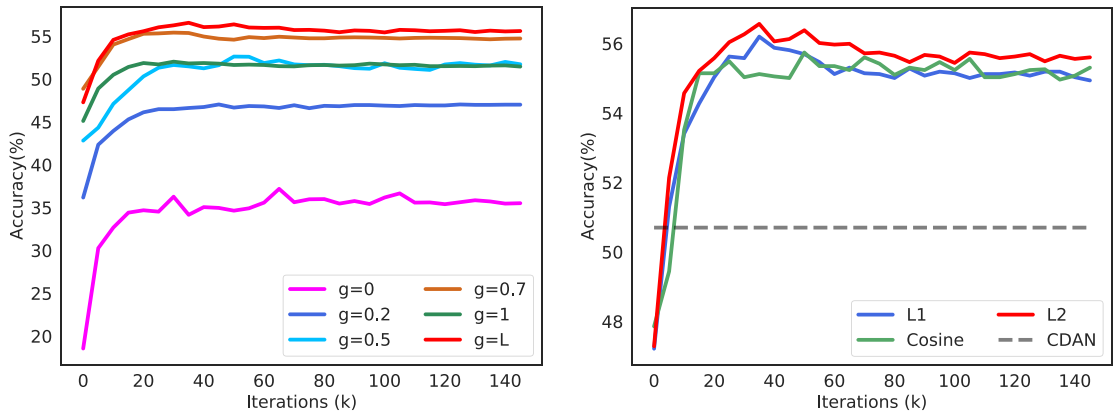
4.6.1. Train and test time comparison

Besides, we also compare our RN with the other normalization modules in the perspective of computation cost, *i.e.*, the time in both training and inference stages. In specific, we report the quantitative values of computation cost of different normalization methods on the UDA task of Pr → Rw (Office-Home) with 4 threads and one Tesla V100 GPU. The backbones are the ResNet-50 by default. To eliminate the noise in the estimations, the training time is calculated based on the average of the time cost of 10,000 iterations, including forward and backward operations. And the test times are the total time cost of evaluating the whole target domain dataset.

These results are listed in the Table 6. We can observe that the training time cost of RN is more than vanilla BN, AutoDIAL, and DSBN, but less than TN. On the other hand, the test time cost of RN is very close to the vanilla BN, and also less than other normalization methods, implying that our RN has more advantages in practical applications. These results demonstrate that our RN achieves better trade-off between the time costs of both train and test phases and simultaneously achieves better performance.

Table 4
Classification accuracies (%) of different normalization methods on three domain adaptation benchmarks for UDA.

Method	ImageCLEF-DA							VisDA-C		Office-Home				
	I→P	P→I	I→C	C→I	C→P	P→C	AVG	ResNet-50	ResNet-101	Ar →X	Cl →X	Pr →X	Rw →X	AVG
DANN(+BN) [1]	75.0	86.0	96.2	87.0	74.3	91.5	85.0	54.9	57.4	58.3	55.5	52.8	63.9	57.6
DANN+AutoDIAL [14]	77.3	88.8	95.3	89.5	**79.0**	91.3	86.9	62.5	64.7	61.4	55.4	54.6	63.9	58.8
DANN+DSBN [16]	77.2	88.2	93.8	90.3	77.8	89.3	86.1	65.0	69.6	57.0	55.2	50.2	56.8	54.8
DANN+TN [15]	**78.2**	89.5	95.5	91.0	76.0	91.5	87.0	66.3	-	58.8	58.3	55.6	64.6	59.3
DANN+RN	78.1	**90.1**	**96.3**	**91.7**	78.0	**94.0**	**88.0**	**71.4**	**74.9**	**61.6**	**63.4**	**59.6**	**67.2**	**63.0**
CDAN(+BN) [9]	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7	70.0	73.9	65.8	65.9	61.9	69.7	65.8
CDAN+AutoDIAL [14]	77.8	90.3	96.8	91.2	77.2	94.5	88.0	71.8	74.5	65.3	66.4	61.8	73.9	67.4
CDAN+DSBN [16]	76.2	92.2	94.9	90.1	74.0	94.3	86.9	72.9	78.6	65.5	65.0	58.1	66.7	64.1
CDAN+TN [15]	78.3	90.8	96.7	92.3	78.0	94.8	88.5	71.4	-	66.3	68.4	64.5	71.3	67.6
CDAN+RN	**78.6**	**92.7**	97.2	**92.8**	**79.1**	**94.8**	**89.2**	**79.6**	**80.1**	**68.8**	**71.7**	**68.4**	**73.4**	**70.6**



(a) (b)

Fig. 4. (a) The influence of g in RA. "L" means g is learnable. (b) Analysis of the measures of correlations strength in RC. The backbone is ResNet-50 and the UDA method is CDAN [9].

Table 6
Comparisons of training and testing time (s) of different feature normalization methods on the same domain adaptation baseline (CDAN + ResNet-50).

Setting	CDAN				
	(+BN)	+ AutoDIAL	+ TN	+ DSBN	+ Ours
Train	0.13	0.38	0.71	0.14	0.50
Test	1.91	3.76	64.40	5.17	1.90

Table 7
Ablation study results of RC and RA-Gate with CDAN [9] (ResNet-50) on Office-Home benchmark.

RC	RA	Ar → X	Cl → X	Pr → X	Rw → X	AVG
-	-	65.8	65.9	61.9	69.7	65.8
-	✓	67.3	69.4	65.5	72.5	68.9
✓	✓	**68.8**	**71.7**	**68.4**	**73.4**	**70.6**

4.7. Ablation study

4.7.1. Ablation study (RC & RA)

To investigate the effects of RC and RA, we conduct additional ablation study experiments on Office-Home as an instance. The results are shown in Table 7, where "→X" means to other 3 domains and we report the average accuracy of the 3 transfer tasks. Based on CDAN+BN refer to the first row, we progressively add the RA and RC, respectively. Noting that RC cannot be trained independently.

It is clear that aggregating statistics of cross-domain corresponding channels (*i.e.*, the second row) outperforms the baseline, and the full method (*i.e.*, the third row) achieves the best performance. It verifies the effectiveness of the exploitation of the correlation of cross-domain non-corresponding channels. Similar observations can be found in other adaptation scenarios. Therefore, indicating the effectiveness and necessity of the RC and RA in our RN.

4.7.2. Influence of g in RA

To investigate the influence of g in RA, we conduct quantitative analysis on the UDA task "Art → Clipart" from Office-Home dataset.

The results are visualized in Fig. 4(a). Noting that "L" means g is learnable. Obviously, the learnable g achieves the best performance. The training curves demonstrate that the best value interval of RA should be 0.5∼1, verifying that the effectiveness of our constraint on the learnable g. It is easy to understand that $g = 0$ and $g = 1$ mean that training models only with outputs of RC and without outputs of RC, respectively, and $g < 0.5$ means that using less the original statistics leads to lose domain-specific information. The results also ensure the effectiveness of RC. To sum up, both the RA and RC are effective and benefit to domain adaptation.

4.8. Further investigation

4.8.1. Analysis of measures of correlations

To explain the importance of l_2 distance in calculating correlations, we compare the popular different distance metrics on correlations measures. The experiments are conducted on the UDA task "Art→Clipart" in Office-Home dataset.

As illustrated in Fig. 4(b), the l_2 distance achieves the best performance because it enables the channels with similar patterns to have larger weights. Additionally, the l_2 obtains the results with small margin ($< 1\%$) than other measures, indicating the RC is robust to different distance measures. Moreover, with different distance measures, our RN consistently obtains significant improvement over the baseline method, indicating the effectiveness of our RN. Similar observations can also be found in other DA scenarios.

4.8.2. Training convergence

To illustrate the convergence performance and training stability of our RN, we present the classification accuracy during training on the UDA task Art → Clipart of Office-Home. The similar training curves are observed in other adaptation scenarios.

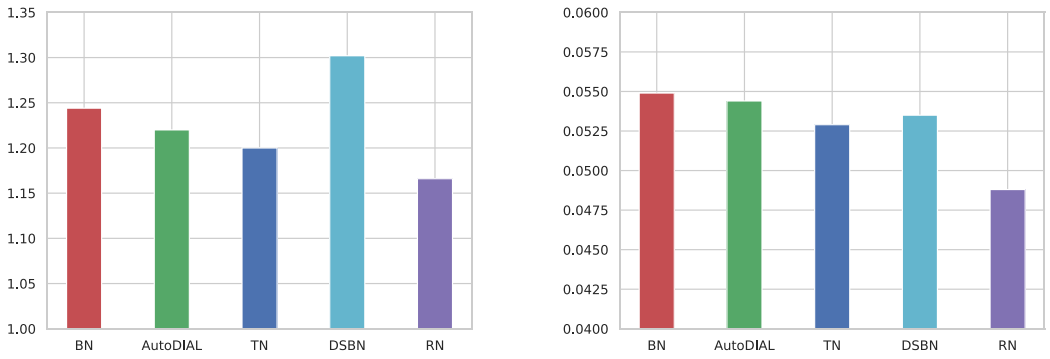
As illustrated in Fig. 5, the proposed RN fast and stably converges to the best accuracy, and achieves the optimal accuracy of over 95% with only 1,000 training iterations (black dotted line), compared with other existing normalization counterparts. We also notice that the accuracy curve of DSBN drops after 60 training iterations, and it indicates that CDAN with DSBN suffers from negative transfer. This observation also verifies the importance of feature normalization module in domain adaptation tasks.

4.9. Theoretical understanding

4.9.1. Theoretical insight

Formally, as Ben-David et al. [67] pioneered, the learning bound of domain adaptation is formulated as follows:

$$\varepsilon_{\mathcal{T}}(h) \leq \varepsilon_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda. \tag{10}$$



(a) A-distance (b) λ

Fig. 5. Training convergence analysis of various normalization techniques when the backbone is ResNet-50 [18] and the UDA baseline method is CDAN [9].

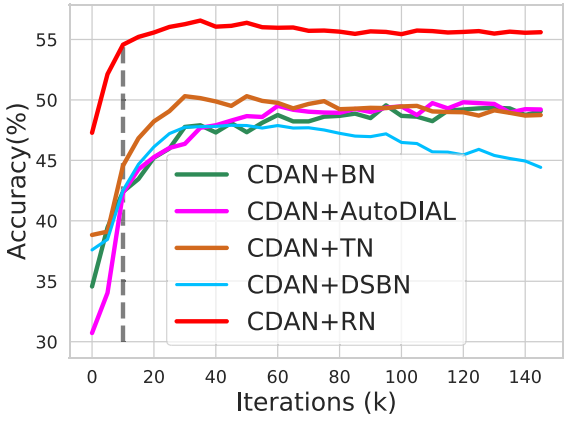


Fig. 6. Theoretical analysis of A-distance (a) and λ in Eq. 10 (b) when using different kinds of feature normalization methods on the same domain adaptation method (*e.g.*, CDAN).

Equation (10) bounds the expected risk $\varepsilon_{\mathcal{T}}(h)$ of a hypothesis h on the target domain by: 1) the expected risk of h on the source domain, $\varepsilon_{\mathcal{S}}(h)$; 2) the A-distance [67], $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T}) = 2(1-2\epsilon)$, a domain-divergence measure, where the ϵ is the error rate of a domain classifier which is trained to discriminate source and target domains; 3) the risk λ of an ideal joint hypothesis h^* for both source and target domains. The A-distance and the λ helps us better understand the rationale of one certain method in the topic of domain adaptation.

To further investigate the theoretical advantage of the proposed RN beyond the existing normalization modules, we estimate the A-distance and the λ on the adaptation task of A $\rightarrow$ C (Office-Home dataset) with CDAN + various normalization methods. The results are shown in Fig. 6(a) and (b) that, Compared with the other normalization counterparts, our RN helps CDAN obtain lower values in both A-distance and λ. It indicates that the proposed RN facilitates more transferable representation from the perspectives of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T})$ and λ. Consequently, when learning visual representation with better transferability, our RN is able to obtain better generalization performance.

4.9.2. Distance of the nearest channels across domains

In Table 8, we calculate the distance of any two channels across domains in the last normalization module in each stage. The four stages denote the four "layer" in ResNet-50 with channel numbers as 256, 512, 1024, and 2048, respectively. The distance is calculated as follows:

$$d^{(j)} = \left| \frac{\mu_s^{(j)}}{\sqrt{\sigma_s^{2(j)}}} - \frac{\mu_t^{(j)}}{\sqrt{\sigma_t^{2(j)}}} \right|, \tag{11}$$

where j denotes the jth channel, which is introduced by Wang et al. [15]. Noting that the source and target domains share the same mean and variance in BN, and we do not calculate the distance across domains. The goal of RN is to find each channel's compensatory information and then aggregate them. The compensatory consists of the information of both the corresponding and non-corresponding channels, where the nearest channels across domains has the largest correlation weight. Hence, we report the sum of the distances between all pairs of the nearest channels. The smaller distance means the greater ability to align both corresponding and non-corresponding channels to a certain extent. We can observe that RN obtains the smallest value, implying the better performance of alignment of RN than other methods. Besides, among the nearest channels, the proportion of corresponding channel is very small, verifying the misalignment between corresponding channels across domains.

Table 8
The distance of the nearest channels across domains in the last normalization module in `stage4` of ResNet-50. "%" denotes the ratio of the corresponding channels in the nearest channels across domains.

Method	Stage				Σ
	1	2	3	4	
CDAN+AutoDIAL	3.77	3.48	1.84	0.97	10.1
	5.1%	3.5%	2.0%	1.4%	-
CDAN+DSBN	5.23	2.25	1.57	0.88	9.9
	3.9%	3.7%	0.8%	0.3%	-
CDAN+TN	4.01	3.43	1.87	0.72	10.0
	5.1%	3.1%	2.8%	0.8%	-
CDAN+Ours	3.06	3.31	1.52	0.62	8.5
	3.1%	5.6%	3.0%	1.4%	-

4.10. Additional visualization

4.10.1. Visualization of g in RA

For better understanding of RA, we also show the visualization of g in RA on UDA tasks Art $\rightarrow$ Clipart of Office-Home. We show

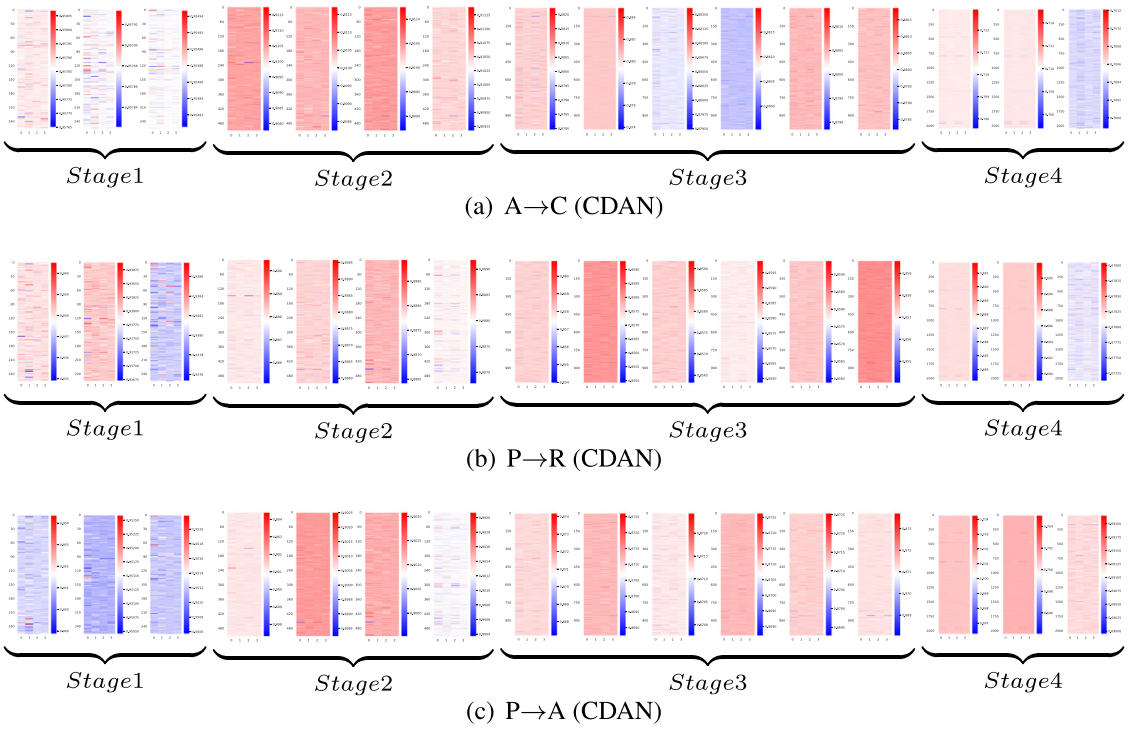


Fig. 7. Heat-map visualization of RA-gates on three randomly selected adaptation tasks from Office-Home benchmarks. We use ResNet-50 as the backbone. These pictures are best viewed in the electronic version.

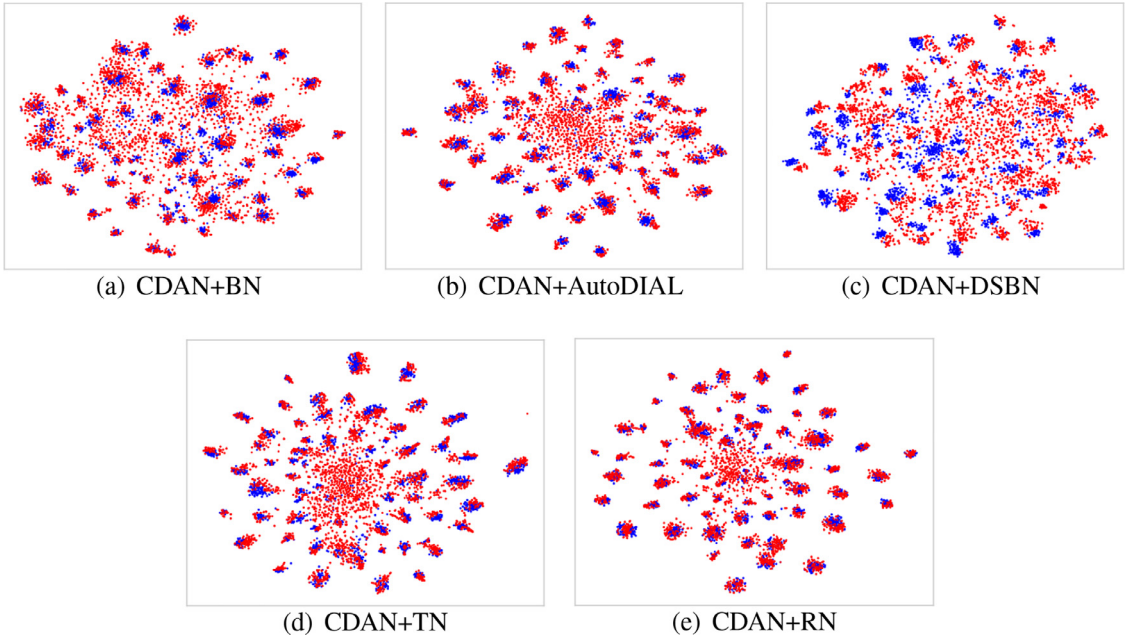


Fig. 8. Visualization of features from the models with different normalization layers on the UDA task Art (Source) → Clipart (Target) from Office-Home benchmark. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the last RN of each layer (*i.e., bn3* in each layer of ResNet50) due to the page limitation. We refer to each stage as stage 1, 2, 3 and 4 with the channel numbers as 256, 512, 1024, and 2048. The "1, 2, 3, 4" on the abscissa axis denote the source mean, source variance, target mean, and target variance, respectively. The ordinates denote the values of g.

As illustrated in Fig. 7, RA conducts the domain alignment at the intermediate layers in different ways automatically. Note that, as the number of channels increases, the weights of gates become smaller, which is consistent with the conclusion that the different transferability in the various layers in [68], and ensures the significance of RA. Similar observations can also be found in other DA scenarios.

4.10.2. Feature visualization

To further understand the effectiveness of the proposed RN, we visualize the learned representation spaces of different feature normalization modules: vanilla BN [17], AutoDIAL [14], DSBN [16],

TN [15], and the proposed RN. Typically, we leverage t-SNE [66] to visualize the feature representations in the bottleneck layer of ResNet-50.

As shown in Fig. 8, we notice that CDAN+DSBN suffers from negative transfer, which is the main reason for the sub-optimal results (see Tables 1, 2, 3, and 5). We also observe that the source and target representations are aligned better by the models integrated RN, compared with existing normalization counterparts. It demonstrates that our RN is effective to learn the domain-invariant information. Meanwhile, the cluster centers of two domains in the same class are closer, indicating that the greater ability of RN to learn the discriminative features.

5. Conclusion

In this paper, we propose a novel normalization layer for domain adaptation, termed Reciprocal Normalization (RN). We devise RN to address the problem that losing the domain information due to the misalignment of channels across domains. The proposed RN structurally aligns the source and target domains by conducting reciprocity across domains. As a generic alternative to BN, our RN can be easily applied to mainstream domain adaptation approaches. Extensive experiments on three benchmarks and three typical adaptation tasks validate that: i) the proposed RN outperforms existing normalization techniques in the context of domain adaptation; ii) popular domain adaptation approaches consistently benefit from our RN and obtain better classification performance on the target domain.

For future work, we will attempt to reveal the theoretical insights within our RN and verify its versatility in other domain adaptation tasks (*e.g.*, object detection, text detection, semantic segmentation, and person re-identification).

Declaration of Competing Interest

None.

Data availability

Data will be made available on request.

References

[1] Y. Ganin, V.S. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International Conference on Machine Learning, 2015.
[2] G. Kang, L. Jiang, Y. Yang, A.G. Hauptmann, Contrastive adaptation network for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[3] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation, in: International Conference on Machine Learning, 2020.
[4] Y. Zuo, H. Yao, C. Xu, Attention-based multi-source domain adaptation, IEEE Trans. Image Process. 30 (2021).
[5] Z. Luo, X. Zhang, S. Lu, S. Yi, Domain consistency regularization for unsupervised multi-source domain adaptive classification, Pattern Recognit. 132 (2022) 108955.
[6] M. Long, H. Zhu, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, in: International Conference on Machine Learning, 2017.
[7] M. Xu, H. Wang, B. Ni, Q. Tian, W. Zhang, Cross-domain detection via graph-induced prototype alignment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[8] W. Zhou, Y. Wang, J. Chu, J. Yang, X. Bai, Y. Xu, Affinity space adaptation for semantic segmentation across domains, IEEE Transactions on Image Processing, IEEE, 2020.
[9] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, Advances in Neural Information Processing Systems, 2018.
[10] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, J. Huang, Progressive feature alignment for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[11] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, K. Kim, Image to image translation for domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[12] F. Pizzati, R.d. Charette, M. Zaccaria, P. Cerri, Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation, WACV, 2020.
[13] Y. Li, N. Wang, J. Shi, J. Liu, X. Hou, Revisiting batch normalization for practical domain adaptation, in: International Conference on Learning Representations, 2017.
[14] F.M. Cariucci, L. Porzi, B. Caputo, E. Ricci, S.R. Bulò, AutoDIAL: automatic domain alignment layers, in: Proceedings of the IEEE International Conference on Computer Vision, 2017.
[15] X. Wang, Y. Jin, M. Long, J. Wang, M.I. Jordan, Transferable normalization: towards improving transferability of deep neural networks, Advances in Neural Information Processing Systems, 2019.
[16] W.-G. Chang, T. You, S. Seo, S. Kwak, B. Han, Domain-specific batch normalization for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[17] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015.
[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.
[19] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: maximizing for domain invariance, arXiv preprint arXiv:1412.3474(2014).
[20] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International Conference on Machine Learning, 2015.
[21] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, A.J. Smola, A kernel method for the two-sample-problem, Advances in Neural Information Processing Systems, 2007.
[22] C.-Y. Lee, T. Batra, M.H. Baig, D. Ulbricht, Sliced Wasserstein discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[23] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, CoRR (2017) abs/1702.05464.
[24] X. Chen, S. Wang, M. Long, J. Wang, Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation, in: International Conference on Machine Learning, 2019.
[25] X. Jiang, Q. Lao, S. Matwin, M. Havaei, Implicit class-conditioned domain alignment for unsupervised domain adaptation, in: International Conference on Machine Learning, 2020.
[26] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, Q. Tian, Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[27] H. Tang, K. Chen, K. Jia, Unsupervised domain adaptation via structurally regularized deep clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[28] M. Mancini, L. Porzi, S. Rota Bulò, B. Caputo, E. Ricci, Boosting domain adaptation by discovering latent domains, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[29] R. Shu, H. Bui, H. Narui, S. Ermon, A DIRT-T approach to unsupervised domain adaptation, in: International Conference on Learning Representations, 2018.
[30] Q. Wang, T. Breckon, Unsupervised domain adaptation via structured prediction based selective pseudo-labeling, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
[31] N. Xiao, L. Zhang, Dynamic weighted learning for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[32] W. Li, S. Chen, Unsupervised domain adaptation with progressive adaptation of subspaces, Pattern Recognit. 132 (2022) 108918.
[33] S. Li, C.H. Liu, Q. Lin, B. Xie, Z. Ding, G. Huang, J. Tang, Domain conditioned adaptation network, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
[34] S. Li, F. Lv, B. Xie, C.H. Liu, J. Liang, C. Qin, Bi-classifier determinacy maximization for unsupervised domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
[35] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, M.-H. Yang, Cross-domain few-shot classification via learned feature-wise transformation, in: International Conference on Learning Representations, 2019.
[36] Y. Du, X. Zhen, L. Shao, C.G. Snoek, MetaNorm: learning to normalize few-shot batches across domains, in: International Conference on Learning Representations, 2020.
[37] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, Advances in Neural Information Processing Systems, 2016.
[38] Y. Wu, K. He, Group normalization, in: Proceedings of the European Conference on Computer Vision, 2018.
[39] P. Luo, J. Ren, Z. Peng, R. Zhang, J. Li, Differentiable learning-to-normalize via switchable normalization, in: International Conference on Learning Representations, 2018.
[40] H. Liu, A. Brock, K. Simonyan, Q. Le, Evolving normalization-activation layers, Advances in Neural Information Processing Systems, 2020.
[41] S.-H. Gao, Q. Han, D. Li, M.-M. Cheng, P. Peng, Representative batch normalization with feature calibration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[42] Y. Li, N. Vasconcelos, Efficient multi-domain learning by covariance normalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[43] S. Roy, A. Siarohin, E. Sangineto, S.R. Bulo, N. Sebe, E. Ricci, Unsupervised domain adaptation using feature-whitening and consensus loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[44] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.
[45] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, VisDA: the visual domain adaptation challenge, arXiv preprint arXiv:1710.06924(2017).
[46] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[47] W. Zhang, W. Ouyang, W. Li, D. Xu, Collaborative and adversarial network for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[48] S. Lee, D. Kim, N. Kim, S.-G. Jeong, Drop to adapt: learning discriminative features for unsupervised domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.
[49] R. Xu, G. Li, J. Yang, L. Lin, Larger norm more transferable: an adaptive feature norm approach for unsupervised domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.
[50] Y. Zou, Z. Yu, X. Liu, B. Kumar, J. Wang, Confidence regularized self-training, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.
[51] V.K. Kurmi, S. Kumar, V.P. Namboodiri, Attending to discriminative certainty for domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[52] Y. Zhang, T. Liu, M. Long, M. Jordan, Bridging theory and algorithm for domain adaptation, in: International Conference on Machine Learning, 2019.
[53] Q. Chen, Y. Liu, Structure-aware feature fusion for unsupervised domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
[54] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, T. Qi, Gradually vanishing bridge for adversarial domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[55] Y. Wu, D. Inkpen, A. El-Roby, Dual mixup regularized learning for adversarial domain adaptation, in: Proceedings of the European Conference on Computer Vision, 2020.
[56] K. Saito, D. Kim, S. Sclaroff, K. Saenko, Universal domain adaptation through self supervision, Adv. Neural Inf. Process. Syst. (2020).
[57] J. Zhang, Z. Ding, W. Li, P. Ogunbona, Importance weighted adversarial nets for partial domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[58] Z. Cao, M. Long, J. Wang, M.I. Jordan, Partial transfer learning with selective adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[59] Z. Cao, K. You, M. Long, J. Wang, Q. Yang, Learning to transfer examples for partial domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[60] J. Liang, Y. Wang, D. Hu, R. He, J. Feng, A balanced and uncertainty-aware approach for partial domain adaptation, in: Proceedings of the European Conference on Computer Vision, 2020.
[61] K. Fatras, T. Séjourné, R. Flamary, N. Courty, Unbalanced minibatch optimal transport; applications to domain adaptation, in: International Conference on Machine Learning, 2021.
[62] B. Sun, K. Saenko, Deep CORAL: correlation alignment for deep domain adaptation, in: Proceedings of the European Conference on Computer Vision, 2016.
[63] D. Li, T. Hospedales, Online meta-learning for multi-source and semi-supervised domain adaptation, in: European Conference on Computer Vision, 2020.
[64] N. Venkat, J.N. Kundu, D. Singh, A. Revanur, V.B. R, Your classifier can secretly suffice multi-source domain adaptation, Advances in Neural Information Processing Systems, 2020.
[65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: an imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems, 2019.
[66] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, JMLR (2008).
[67] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. (2010).
[68] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? Advances in Neural Information Processing Systems, 2014.

Zhiyong Huang He received his BEng degree in Control and Computer Engineering from North China Electric Power University in July 2021. He received his BEng degree from North China Electric Power University in 2018. His research interest include s single image super resolution and domain adaptation.

Kekai Sheng He received his PhD degree from National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences in 2019. He received his BEng degree in Telecommunication Engineering from University of Science and Technology Beijing in 2014. He is currently a researcher engineer at Youtu Lab, Tencent Inc. His research interests in clude image quality evaluation, domain adaptation, and AutoML.

Ke Li He received the BEng degree in Computer Science from Xiamen University, Fujian, China, in July 2018. He is currently a research engineer at Youtu Lab, Tencent Inc. His research interests involve self supervised learning, deep learning, and machine learning.

Jian Liang He received the BE degree in Electronic Information and Technology from Xi'an Jiaotong University and PhD degree in Pattern Recognition and Intelligent Systems from National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences in July 2013, and January 2019, respectively. He was a research fellow at National University of Singapore from June 2019 to April 2021. Now he joins NLPR and works as an associated professor. His research interests focus on transfer learning, pattern re cognition, and computer vision.

Taiping Yao He received the BEng degree in Electrical Engineering from Shanghai Jiao Tong University, Shanghai, China, in July 2019. He is currently a researcher engineer at Youtu Lab, Tencent Inc. His research interests involve computer vision and deep learning.

Weiming Dong He is a Professor in the Sino European Lab in Computer Science, Automation and Applied Mathematics (LI AMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences. He received his BSc and MSc degrees in Computer Science in 2001 and 2004, both from Tsinghua University, China. He received his PhD in Computer Science from the University of Lorraine, France, in 2007. His research interests include visual media synthesis and image recognition. Weiming Dong is a member of the ACM and IEEE.

Dengwen Zhou He is a Professor in the School of Control and Computer Engineering, North China Electric Power University, Beijing, China. He has long been engaged in research on image processing, including image de noising, image demosaicking, image interpolation and image super resolution etc. Current research focuses on the applications based on neural networks and deep learning in image processing and computer vision.

Xing Sun He is currently a team lead and senior researcher in Youtu Lab, Tencent Inc. Before that, he received his PhD degree under the supervision of Prof. Edmund Y. Lam in Imaging Systems Laboratory, and Dr. Nelson Yung in Laboratory for Intelligent Transportation Systems Research in the Department of Electrical and Electronic Engineering at The University of Hong Kong in 2016. He received his BS degree at Nanjing University of Science and Technology in Jun. 2012.