Collaborative Normalization for Unsupervised Domain Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Batch Normalization (BN) as an important component assists Deep Neural Networks achieving promising performance for extensive learning tasks by scaling distribution of feature representations within mini-batches. However, the application of BN suffers from performance degradation under the scenario of Unsupervised Domain Adaptation (UDA), since the estimated statistics fail to concurrently describe two different domains. In this paper, we develop a novel normalization technique, named Collaborative Normalization (CoN), for eliminating domain discrepancy and accelerating the model training of neural networks for UDA. Unlike typical strategies only exploiting domain-specific statistics during normalization, our CoN excavates cross-domain knowledge and simultaneously scales features from various domains by mimicking the merits of collaborative representation. Our CoN can be easily plugged into popular neural network backbones for cross-domain learning. One the one hand, **theoretical analysis** guarantees that models with CoN promote discriminability of feature representations and accelerate convergence rate; on the other hand, **empirical study** verifies that replacing BN with CoN in popular network backbones effectively improves classification accuracy with **4%** in most learning tasks across three cross-domain visual benchmarks.

1 Introduction

The hierarchical structure of Deep Neural Networks (DNN) facilitates itself to achieve appealing performances with prolific semantic representations in most learning tasks (Saito et al. (2019); Xu et al. (2019)). As an indispensable component in DNN, batch normalization (BN) aims to scale internal features to promote modeling ability of DNN (Ioffe & Szegedy (2015)). Concretely, typical BN preserves the scale of distribution invariant among various network layers by normalizing features, avoiding gradient vanishing and accelerating model convergence (Li et al. (2018)). To accurately estimate property of distribution, the training of DNN thus requires abundant well-labeled instances, which is unsuitable for real-life scenarios.

Unsupervised Domain Adaptation (UDA) casts a light on such a barren condition and explores external source domain with sufficient annotation to build a model generalized to unlabeled target domain. The primary challenge for UDA is to overcome domain shift that multi-domains belong to various distributions (Zhang et al. (2019a); Ma et al. (2019)). Existing mainstream solutions attempt to eliminate cross-domain discrepancy by learning domain-invariant representations with DNN (Zhang et al. (2019b); Long et al. (2017)). Along this line, one successful strategy adopts adversarial mechanism between feature extractor and domain discriminator to perform domain confusion (Liu et al. (2019); Zhang et al. (2019b)). Other efforts (Long et al. (2017); Kang et al. (2019)) focusing on the alignment of various distributions expect both domains to share the identical statistics (*e.g.*, mean value and co-variance). For the convenient implementation, these works generally bind the corresponding constraints with full-connection features following convolutional operations. However, the effect of objective function flowing in stacked network architecture gradually becomes too weak to align source and target features specially on shallow layers during back propagation.

To overcome such a problem, the variants of BN linking adjacent layers attract massive attentions on domain adaptation. Traditional methods typically adopt domain-specific BNs to separately scale source and target features (Kang et al. (2019); Long et al. (2018)), however, their major drawback lies in the difficulty of capturing cross-domain association. From statistical perspective, adaptive batch normalization (AdaBN) (Li et al. (2018)) thus exploits the same BN module across various domains by training normalization component on source domain and fixing parameters to identify target samples for inference stage. However, significant domain discrepancy has negative influence on the direct application of source statistics on target domain. To mitigate collision, the automatic domain alignment layer (Cariucci et al. (2017)) considers the linear combination of source and target statistics as indicator of BN. The primary challenge is accurately to select the combination coefficient as the key to succeed. Another perspective delves into the learning of transferable feature representations. Specifically, TransNorm (TN) claims that the similar convolutional channels across both domains tend to record similar patterns intensified in BN operation to promote the transferability of features (Chen et al. (2019)). For visual signals, theses attributions, however, are corresponding to the same concepts such as blue sky, green grass *et.al* instead of our interested objects. Therefore, the enhancement of them brings a little benefit for classification of target domain.

Different from their viewpoints, we explore cross-domain feature alignment during forward propagation from manifold distribution perspective (Luo et al. (2020); Fernando et al. (2013)). Although convolutional representations distribute in high-dimensional feature space, instances from the identical category lie in the same cluster within each domain. However, domain divergence results in constituting various subspaces of source and target features with the same annotation. In this paper, we have alternatively transformed domain adaptation task into subspace fusion problem. Thus, we propose a novel collaborative normalization (CoN) to answer how to carry out cross-domain subspace alignment in forward propagation of features. First, CoN module exploits domain-specific statistics to normalize features to avoid destroying original data distribution. Second, CoN investigates cross-domain structural information through the global pooling of convolutional features. Finally, CoN attempts to estimate the location of source (target) features in target (source) subspace and gradually align samples from its own subspace to the other. The main contributions of our work are summarized in three folds:

- We advance traditional BN with a novel feature adjustment mechanism in forward propagation and easily plug our CoN module into convolutional layers without additional parameters.
- Our theoretical analysis further illustrates why our CoN effectively achieves domain alignment via translation between source and target samples and accelerates convergence speed.
- Experimental evaluations on several visual domain adaptation benchmarks demonstrate that our CoN facilitates convolutional neural network to learn better domain-invariant feature representations than other normalization techniques as traditional BN.

2 Related Work

In this section, we mainly review unsupervised domain adaptation problem and batch normalization strategies, and highlight the difference between our proposed method.

Unsupervised Domain Adaptation (UDA) aims to train a source-supervised model with high generalization on target domain. The primary challenges for UDA are to learn transferable feature and achieve the alignment of distribution. To overcome such issues, domain-adversarial manner is adopted to train a neural network with generator and discriminator and learn domain-invariant features (Chen et al. (2019); Zhang et al. (2019b)). Another solution claims that statistics of data reflect the situation of distribution and forces source and target domains to share the identical indicators such as MMD and its variants (Long et al. (2015; 2017); Kang et al. (2019)). Both schemes apply back propagation to delivery the corresponding constraints by using objective function on top layers. However, gradient vanishing as the increasing number of network layers gradually decrease the effect of condition on bottom layers. **Unlike them**, this paper attempts to eliminate cross-domain discrepancy during forward propagation by proposing a novel network normalization component.

Batch Normalization (BN) as an important component has been widely studied to demonstrate that it effectively promotes the performance of DNN by scaling internal representations across network layers (He et al. (2016)). There exist many variants of BN to satisfy specific requirement for other applications (Cooijmans et al. (2016); Wang et al. (2018); Nam & Kim (2018)). To fight off domain mismatch, a few works thus explore novel techniques based on domain-specific BN to address issue of domain adaptation (Li et al. (2018); Wang et al. (2019b); Roy et al. (2019)). AutoDIAL attempts to construct new statistics through the linear combination of indicators derived from source and target domains to concurrently normalize all features (Cariucci et al. (2017)). However, the optimal parameters of combination are yet not simply accessible. DSBN Chang et al. (2019) adopts domain-specific BN to scale source and target features to preserve more domain-specific knowledge, which difficultly captures the cross-domain association to achieve distribution alignment. Furthermore, TN points out several channels of convolutional features are more likely to record similar content for both domains and intensifies representation of these channels to promote the transferability of features (Wang et al. (2019a)). The sense of TN is that these similar contents are task-relevant patterns such as objects instead of background, which is difficult to guarantee. **Differently**, the proposed method considers the translation of sample from one domain to another and carries such motivation into normalization. Such a strategy not only overcomes domain shift, but also promotes the discriminability of features due to advantage of collaborative representation.

3 The Proposed Approach

3.1 Preliminaries and Motivation

Denote a well-labeled source domain $\mathcal{D}_s = \{(\mathbf{X}_i^s, \mathbf{l}_i^s)\}_{i=1}^{n_s}$ and a target domain $\mathcal{D}_t = \{\mathbf{X}_i^t\}_{i=1}^{n_t}$ without any annotation, where $\mathbf{X}_i^{s/t}$ represents visual signal from the corresponding domain and $\mathbf{l}_i^s \in \mathbb{R}^{C\times 1}$ is the label for $\mathbf{X}_i^s$, where C is the number of category. Unsupervised Domain Adaptation (UDA) aims to borrow knowledge from $\mathcal{D}_s$ to annotate the unlabeled target instances. Benefiting from prolific semantic knowledge learned by hierarchical network architecture, existing explorations for UDA apply DNN to generate domain-invariant features (Zhang et al. (2019b); Tang & Jia (2020)). Without loss of generality, the convolutional features of the k-th hidden layer are defined as $\mathbf{F}^{s/t} = \{\mathbf{F}_i^{s/t} \in \mathbb{R}^{L\times W\times H} | i = 1, 2, \cdots, m\}$ for each mini-batch with m samples, where L, W and H mean the length, width and channel number of feature tensor, respectively.

Domain-specific batch normalizations following convolutional operation are explored to scale representations $\mathbf{F}^s$ and $\mathbf{F}^t$ as Figure 1 (a) in green background with the estimated mean value and co-variance of $\mathbf{F}_{(j)}^{s/t}$ for the j-th channel ($j \in \{1, 2, \cdots, H\}$):

$$\mu_{(j)}^{s/t} \leftarrow \frac{1}{mLW}\sum_{i=1}^{m}\sum_{a=1}^{L}\sum_{b=1}^{W}\mathbf{F}_{i,a,b(j)}^{s/t}, \qquad \sigma_{(j)}^{s/t} \leftarrow \frac{1}{mLW}\sum_{i=1}^{m}\sum_{a=1}^{L}\sum_{b=1}^{W}\left(\mathbf{F}_{i,a,b(j)}^{s/t} - \mu_{(j)}^{s/t}\right)^2. \quad (1)$$

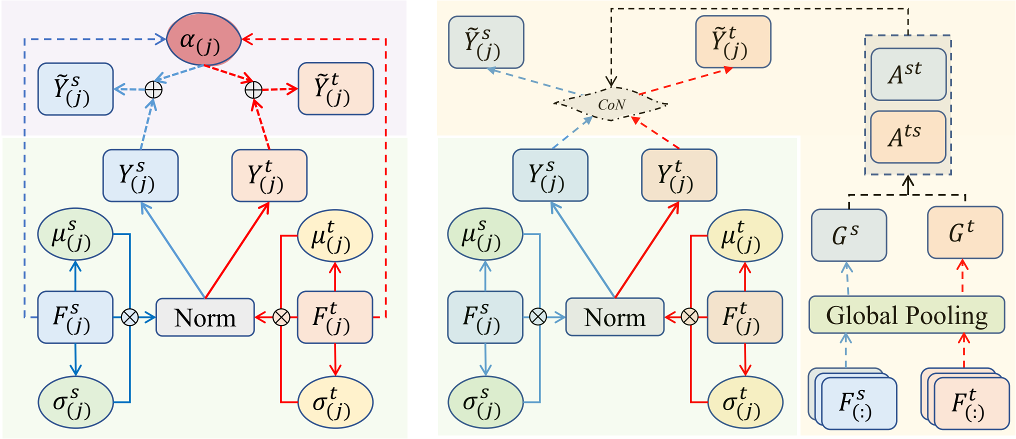
To improve the modeling capacity of neural network, the transformed representation $\hat{\mathbf{F}}_{(j)}^{s/t}$ from $\mathbf{F}_{(j)}^{s/t}$ is further scaled and shifted into the following formulation:

$$\hat{\mathbf{F}}_{i(j)}^{s/t} = \frac{\mathbf{F}_{i(j)}^{s/t} - \mu_{(j)}^{s/t}}{\sqrt{\sigma_{(j)}^{s/t}}}, \qquad \mathbf{Y}_{i(j)}^{s/t} = \gamma_{(j)}\hat{\mathbf{F}}_{i(j)}^{s/t} + \beta_{(j)}, \quad (2)$$

where $\gamma_{(j)}$ and $\beta_{(j)}$ are learnable parameters. Actually, this operation for each domain effectively scales the feature representations across various network layers to stabilize the model training and accelerate the convergence rate. However, such a normalization strategy using different statistics to scale source and target samples suffers from the difficulty of eliminating domain discrepancy. To handle this bottleneck, TN (Wang et al. (2019a)) concerns on learning transferable features and advances typical BN with dashed lines in Figure 1 (a) by sharing information of channel across both domains during forward propagation stage. Concretely, for each channel, TN module computes cross-domain difference $d_{(j)}$ and evaluates the channel transferability via parameter $\alpha_{(j)}$:

$$d_{(j)} = \left|\frac{\mu_{(j)}^s}{\sqrt{\sigma_{(j)}^s}} - \frac{\mu_{(j)}^t}{\sqrt{\sigma_{(j)}^t}}\right|, \qquad \alpha_{(j)} = \frac{H_k(1 + d_{(j)})^{-1}}{\sum_{i=1}^{H_k}(1 + d_{(j)})^{-1}}. \quad (3)$$

According to the above definition, $\alpha_{(j)}$ with large value indicates the j-th channels corresponding to two domains contain similar pattern. Promoting the importance of such channels with $\widetilde{\mathbf{Y}}_i^{s/t} = (1 + \boldsymbol{\alpha}) \odot \mathbf{Y}_i^{s/t}$ to some extent reduces domain shift ($\odot$ denotes element-wise multiplication), where $\boldsymbol{\alpha}$ is a vector with a concatenation of the values $\{\alpha_{(j)}|_{j=1,2,\cdots,H}\}$. TN assumes that these enhanced features include knowledge of our interested object. Unfortunately, the current version fails to



(a) BN & TN

(b) Collaborative Normalization (CoN)

Figure 1: Normalization tools for UDA. (a) Typical Batch Normalization (BN) in green background firstly estimates the mean $\mu^{s/t}_{(j)}$ and variance $\sigma^{s/t}_{(j)}$ from the j-th feature map, and then normalize $F^{s/t}_{(j)}$ into $Y^{s/t}_{(j)}$. TN further extended BN by exploring $\alpha_{(j)}$ (dash lines) to enhance the transferability of several channels. (b) Our CoN advances the BN with dashed lines and borrows cross-domain knowledge derived from all feature maps $F^{s/t}_{(:)}$ to implement collaborative normalization and eliminate domain shift.

guarantee such requirement. For example, even though we discover several similar cross-domain channels extracting background content from visual signals, it is still meaningless to continue reducing difference between them for classification task.

Different from them, we focus on the manifold distribution of features in forward propagation. For each domain, similar feature representations with the same annotation lie in the identical subspace. However, source and target features from the same category distribute in different subspaces due to considerable domain shift. Alternatively, the reduction of domain discrepancy also means the subspace alignment. Motivated by such a consideration, we expect to estimate the location of source (target) features in target (source) domain and gradually align instances from its own subspace to the other. Therefore, this paper proposes a novel collaborative normalization (CoN) strategy to implement our purpose.

3.2 Collaborative Normalization (CoN)

The CoN module mainly involves three operations: domain-specific normalization, collaborative translation and excavation of cross-domain structural knowledge.

Domain Specific Normalization. Domain shift means that source and target instances come from two completely-different distributions. Thus, the normalization of features across both domain with the same statistic easily undermines the original distribution information. To avoid such problem, we follow the usual solution (Ganin et al. (2017)) to scale features with domain-specific statistics and shift them with the identical parameters $(\gamma_{(j)}, \beta_{(j)})$ into $\mathbf{Y}^{s/t}_{i(j)}$.

Collaborative Translation. The core of CoN is to achieve the alignment of source and target subspaces by moving instances from its own subspace to the other. Before the specific implementation, we have to post two important questions: where is the location of samples in another subspace and how to move it into the position. ***The solution to the first challenge*** is motivated by manifold theory that each sample can be represented by the linear combination of others. Without losing generality, we take source-to-target translation as an example and consider the linear combination of target samples as the location of the given source instance in target subspace. For the clarity of illustration, we firstly reshape hidden feature $\mathbf{Y}^{s/t}_{i(j)}$ corresponding to each sample into vector form $\mathbf{y}^{s/t}_i \in \mathbb{R}^{1\times d}$ without (j), where d is the dimension of feature. Given source sample $\mathbf{y}^s_i$, target samples $\mathbf{Z}^t = [(\mathbf{y}^t_1)^\top, (\mathbf{y}^t_2)^\top, \cdots, (\mathbf{y}^t_{m_t})^\top]^\top$ ($\mathbf{y}^t_j \in \mathbb{R}^{1\times d}$) are regarded as a set of basis vectors to represent it, i.e., $\mathbf{y}^s_i \approx \mathbf{m}^t_i \mathbf{Z}^t$, where $\mathbf{m}^t_i \in \mathbb{R}^{1\times m_t}$ denotes coefficient of linear combination for i-th source instance and m_t denotes batch size. Specifically, suppose that $\mathbf{y}^s_i$ and $\mathbf{y}^t_j$ belong to the same category, we should emphasize the contribution of $\mathbf{y}_j^t$ for linear representation, which means that the j-th element in $\mathbf{m}_i^t$ tends to be larger value than others. Similarly, when these two domains change their roles in linear combination, the formulation of collaborative translation will be maintained, i.e., $\mathbf{y}_i^t \approx \mathbf{m}_i^s \mathbf{Z}^s$, where $\mathbf{m}_i^s$ has the same meaning with $\mathbf{m}_i^t$.

With respect to the second challenge, we are motivated by the explanation that $\mathbf{m}_i^t \mathbf{Z}^t$ ($\mathbf{m}_i^s \mathbf{Z}^s$) serves as projection of $\mathbf{y}_i^s$ ($\mathbf{y}_i^t$) on subspace spanned by $\mathbf{Z}^t$ ($\mathbf{Z}^s$) and attempt to adjust source feature to the approximation to achieve subspace alignment. However, we further concern about another question about the reliability of adjustment. Alternatively, when there exists small difference between them, performing the corresponding adjustment tends to be confident, vice versa. Thus, we define adjustment coefficient to evaluate the difference between $\mathbf{y}_i^s$ and $\mathbf{m}_i^t \mathbf{Z}^t$ to gradually conduct adjustment:

$$\eta_i^s = \frac{d}{\|\mathbf{y}_i^s - \mathbf{m}_i^t \mathbf{Z}^t\|_2}, \qquad \eta_i^t = \frac{d}{\|\mathbf{y}_i^t - \mathbf{m}_i^s \mathbf{Z}^s\|_2}, \tag{4}$$

where η_i^s (η_i^t) with larger value means the higher credibility of this adjustment. Therefore, the collaborative translations is formulated as: $\widetilde{\mathbf{Y}}^s = \mathbf{Y}^s + \boldsymbol{\eta}^s \odot (\mathcal{A}^{st}\mathbf{Z}^t - \mathbf{Y}^s)$ and $\widetilde{\mathbf{Y}}^t = \mathbf{Y}^t + \boldsymbol{\eta}^t \odot (\mathcal{A}^{ts}\mathbf{Z}^s - \mathbf{Y}^t)$, where $\boldsymbol{\eta}^{s/t} = [\eta_1^{s/t}, \eta_2^{s/t}, \cdots, \eta_{m_{s/t}}^{s/t}]^\top$, $\mathcal{A}^{st} = [(\mathbf{m}_1^t)^\top, (\mathbf{m}_2^t)^\top, \cdots, (\mathbf{m}_{m_s}^t)^\top]^\top$ and $\mathcal{A}^{ts} = [(\mathbf{m}_1^s)^\top, (\mathbf{m}_2^s)^\top, \cdots, (\mathbf{m}_{m_t}^s)^\top]^\top$. The final step is to reshape $\widetilde{\mathbf{Y}}_i^{s/t}$ into a 2-D feature map with the same size with $\mathbf{Y}_{i(j)}^{s/t}$.

Structural Knowledge Excavation: The next discussion is about the design of transfer coefficient $\mathcal{A}^{st}$ and $\mathcal{A}^{ts}$ in Figure 1 (b). Due to the accessibility of source and target features, we learn the closed-form solution of coefficient via the optimization of the ordinary least square between source and the combined features. However, the strategy with optimization operation postpones the forward propagation of features and hardly captures sample-to-sample relationship without sufficient training instances in each mini-batch. To fight off these drawbacks, we alternatively turn to the application of cross-domain structural knowledge derived from $\mathbf{F}^s$ and $\mathbf{F}^t$. Concretely, when channel number of feature $H \geq 2$, the global pooling operation calculates the average of all elements in each 2-D tensor $\mathbf{F}_{i(j)}^{s/t}$, and then compresses feature of each sample $\mathbf{F}_i^{s/t}$ into $\mathbf{G}_i^{s/t} \in \mathbb{R}^{1\times H}$. Thus, we formulate the element of $\mathcal{A}^{st}$ as $\mathcal{A}_{ij}^{st} = \frac{\mathbf{G}_i^s (\mathbf{G}_j^t)^\top}{\|\mathbf{G}_i^s\|_2 \cdot \|\mathbf{G}_j^t\|_2}$ and $\mathcal{A}^{ts} = (\mathcal{A}^{st})^\top$. For full-connection (FC) layers, the cross-domain graph is directly computed from the cosine distance of source and target features. To this end, we easily plug our CoN layer into any network layers without additional parameters.

3.3 Why can CoN work for UDA?

The subspace spanned by target samples $\mathbf{Z}^t$ is formulated as Φ geometrically shown as a plane in Figure 2. Since any source sample $\mathbf{y}^s$ can be linearly represented by $\mathbf{Z}^t$, we formulate the approximated error between them as $\epsilon = \mathbf{y}^s - \overline{\mathbf{y}}^s$, where $\overline{\mathbf{y}}^s = \mathcal{A}_i^{st}\mathbf{Z}^t$, $\mathcal{A}_i^{st} \in \mathrm{R}^{1\times m_t}$. Meanwhile, we notice that target samples of the i-th category construct a subspace $\Phi_i \in \Phi$ while others form other subspace $\overline{\Phi}_i = \cup_{j=1\&j\neq i}\Phi_j \in \Phi$. In this way, the estimated vector $\overline{\mathbf{y}}^s$ can be decomposed into two components: $\chi_i \in \Phi_i$ and $\overline{\chi}_i \in \overline{\Phi}_i$. Akin to such decomposition, one component of the approximated error is formulated as vector ϵ_i. In the following, we firstly provide theoretical analysis about our CoN and then illustrate how it does work for UDA.

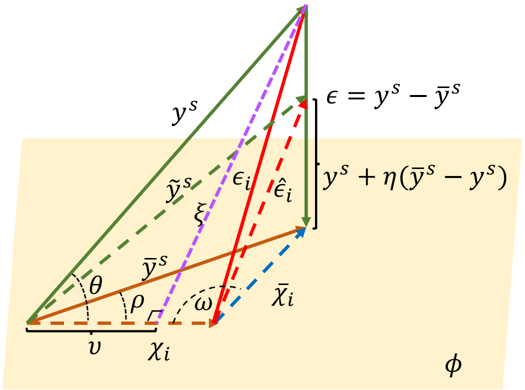


Figure 2: Geometric illustration for the working mechanism of Collaborative Normalization (CoN).

Theorem 3.1. Given target samples divided into C categories and any source sample $\mathbf{y}^s$, we assume each class has only one sample in target domain. For $i, j \in \{1, 2, ..., C\}$, where $i \neq j$, the following conclusion holds: $\nexists \epsilon_j$ such that $\|\epsilon_j\|_2 \leq \|\epsilon_i\|_2$, while $\mathbf{y}^s$ belongs to the i-th category.

Proof. According to the design of CoN, it is straightforward to obtain that $\chi_i = \cos\theta \cdot \mathbf{y}_i^t$ and $\|\chi_i\|_2 = \cos\theta \|\mathbf{y}_i^t\|_2$. In addition, we have to point out that $\epsilon \perp \Phi$. Due to the auxiliary line $\xi \perp \chi_i$, we have:

$$\|\nu\|_2 = \|\mathbf{y}^s\|_2 \cdot \cos\theta = \|\overline{\mathbf{y}}^s\|_2 \cdot \cos\rho, \qquad \cos\rho = \frac{\|\mathbf{y}^s\|_2 \cos\theta}{\|\overline{\mathbf{y}}^s\|_2}. \tag{5}$$

From Figure 2, the formulation $\|\overline{\mathbf{y}}^s\|_2 \cdot \sin\rho = \|\overline{\chi}_i\|_2 \cdot \sin\omega$ holds under the law of sines. In terms of these discussions, we further have the following formulation:

$$\|\overline{\chi}_i\|_2 = \sqrt{\|\overline{\mathbf{y}}^s\|_2 + \|\mathbf{y}_i^t\|_2 \cdot \cos^2\theta - 2\|\overline{\mathbf{y}}^s\|_2 \cdot \|\mathbf{y}_i^t\|_2 \cdot \cos\theta \cdot \cos\rho} = \sqrt{\|\overline{\mathbf{y}}^s\|_2 - \cos^2\theta}. \tag{6}$$

To achieve the final result, we simultaneously shrink vector $\mathbf{y}^s$ and $\mathbf{y}_i^t$ into the same scale, i.e., $\|\mathbf{y}^s\|_2 = \|\mathbf{y}_i^t\|_2 = 1$. From another perspective, we achieve $\|\epsilon_i\|_2^2 = \|\epsilon\|_2^2 + \|\overline{\chi}_i\|_2^2$ and rewrite it as:

$$\|\epsilon_i\|_2^2 = \|\epsilon\|_2^2 + \|\overline{\mathbf{y}}^s\|_2^2 - \cos^2\theta. \tag{7}$$

In terms of Eq. (7), the approximated error $\|\epsilon_i\|_2$ depends on three terms ϵ, $\overline{\mathbf{y}}^s$ and $\cos\theta$. From manifold perspective, features from the same category lie in a very compact subspace. Thus, the cosine similarity between $\mathbf{y}^s$ and $\mathbf{y}_i^t$ both from the i-th category becomes higher than the distance between $\mathbf{y}^s$ and $\mathbf{y}_j^t$ from various classes. With the aid of manifold theory, the conclusion $\nexists \epsilon_j$, $\|\epsilon_j\|_2 \leq \|\epsilon_i\|_2$ holds. Next, we utilize Eq. (7) to illustrate why our CoN can achieve domain alignment. According to CoN module $\widetilde{\mathbf{y}}^s = \mathbf{y}^s + \eta^s(\overline{\mathbf{y}}^s - \mathbf{y}^s)$, $\mathbf{y}^s$ is transformed into $\widetilde{\mathbf{y}}^s$ and the component of approximated error ϵ_i tends to be $\hat{\epsilon}_i$, where $\|\hat{\epsilon}_i\|_2^2 = \|\epsilon - \delta\|_2^2 + \|\overline{\mathbf{y}}^s\|_2^2 - \cos^2\theta$ and $\delta = \eta(\overline{\mathbf{y}}^s - \mathbf{y}^s)$. Therefore, we have the following inequality over ϵ_i and $\hat{\epsilon}_i$:

$$\|\hat{\epsilon}_i\|_2^2 = \|\epsilon\|_2^2 + \|\delta\|_2^2 - 2\|\epsilon\|_2\|\delta\|_2 + \|\overline{\mathbf{y}}^s\|_2^2 - \cos^2\theta \leq \|\epsilon_i\|_2^2, \quad \|\delta\|_2 \leq \|\epsilon\|_2. \tag{8}$$

The above formulation denotes the adjusted features $\widetilde{\mathbf{y}}^s$ is closer to its corresponding target category when compared with the original feature $\mathbf{y}^s$. Therefore, our method utilizes such a reliable adjustment to gradually achieve distribution alignment and improve the discriminative ability of features.

4 Experiments

To verify the effectiveness of Collaborative Normalization (CoN), we apply the proposed method into two well-known deep transfer learning backbones **CDAN** (Long et al. (2018)) and **DANN** (Ganin et al. (2017)) and evaluate their performance on three popular benchmark datasets.

4.1 Experimental Setting

Datasets: *1) Image-CLEF* collects visual signals from three subsets: Caltech-256 (**C**), ImageNet ILSVRC 2012 (**I**) and Pascal VOC 2012 (**P**) with the same number of samples. Concretely, arbitrary subset includes 600 images evenly distributed in 12 categories. *2) Office-31* (Saenko et al. (2010)) as a benchmark dataset of domain adaptation involves 4,652 images from 31 categories. These instances are divided into three subsets: Amazon (**A**, 2,817 images), DSLR (**D**, 498 images) and Webcam (**W**, 795 images). *3) Office-Home* (Venkateswara et al. (2017)) consists of four subsets: Artistic images (**Ar**), Clip Art (**Cl**), Product images (**Pr**) and Real-World images (**Rw**). Four subsets with 15,500 images share the identical label space of 65 categories.

Competitive baselines: We not only explore the state-of-the-art domain adaptation methods including **DAN** (Long et al. (2015)), **JAN** (Long et al. (2017)), **MADA** (Pei et al. (2018)), **DSR** (Cai et al. (2019)), **SymNets** (Zhang et al. (2019b)), **TADA** (Wang et al. (2019b)), **SAFN** (Xu et al. (2019)), **DRMEA** (Luo et al. (2020)), **DADA** (Tang et al. (2020)) but also combine network backbones (CDAN and DANN) with multi-norm strategies: BN (Long et al. (2018)) and TN (Wang et al. (2019a)) as baselines. We follow the standard protocols operated with CDAN and DANN to evaluate the effectiveness of our proposed normalization technique. To make fair comparisons, results of all above methods are directly copied from the corresponding literature under the exactly same protocols.

Experimental results on Image-CLEF, Office-31 and Office-Home datasets are summarized in Tables 1, 2, and 3, respectively. From these results, we achieve three main conclusions. First, the integration of CDAN and CoN surpasses all comparisons in most domain adaptation tasks.

Table 1: Classification Accuracy (%) on Image-CLEF dataset (ResNet-50). The best results among all methods are shown with **underline** while the highest accuracy of three normalization tools is in **Red** type. And BN, TN and CoN are plugged into the same backbone **CDAN**.

Method	Res-Net	JAN	DAN	MADA	SAFN	DRMEA	**BN**	**TN**	**CoN**
I→P	74.8±0.3	76.8±0.4	74.5±0.4	75.0±0.3	79.3±0.1	80.7	77.7	78.3	80.2
P→I	83.9±0.1	88.0±0.2	82.2±0.2	87.9±0.2	93.3±0.4	92.5	90.7	90.8	93.3
I→C	91.5±0.3	94.7±0.2	92.8±0.2	96.0±0.3	96.3±0.4	97.2	97.7	96.7	97.5
C→I	78.0±0.2	89.5±0.3	86.3±0.4	88.8±0.3	91.7±0.0	90.5	91.3	92.3	94.3
C→P	65.5±0.3	74.2±0.3	69.2±0.4	75.2±0.2	77.6±0.1	77.7	74.2	78.0	80.4
P→C	91.2±0.3	91.7±0.3	89.8±0.4	92.2±0.3	95.3±0.1	96.1	94.3	94.3	96.2
Avg	80.7	85.8	82.5	85.8	88.9	89.1	87.7	88.5	90.3

Table 2: Classification Accuracy (%) on Office-31 dataset (ResNet-50). The best results among all methods are shown with **underline** while the highest accuracy of three normalization tools is in **Red** type. And BN, TN and CoN are plugged into the same backbone **CDAN**.

Method	Res-Net	JAN	DADA	SymNets	TADA	SAFN	**BN**	**TN**	**CoN**
A →W	68.4± 0.2	85.4± 0.3	92.3± 0.1	90.8± 0.1	94.3± 0.3	90.3	94.1	95.7	96.4
D→W	96.7±0.1	97.4±0.2	99.2±0.1	98.8±0.3	98.7±0.1	98.7	98.6	98.7	98.4
W→D	99.3±0.1	99.8±0.2	100±0.0	100.0±0.0	99.8±0.2	100.0	100.0	100.0	100.0
A→D	68.9±0.2	84.7±0.3	93.9±0.2	93.9±0.5	91.6±0.3	90.7	92.9	94.0	96.0
D→A	62.5±0.3	68.6±0.3	74.4±0.1	74.6±0.6	72.9±0.2	73.4	71.0	73.4	77.3
W→A	60.7±0.3	70.0±0.4	74.2±0.1	72.5±0.5	73.0±0.3	71.2	69.3	74.2	75.7
Avg	76.1	84.3	89.0	88.4	88.4	87.6	87.7	89.3	90.6

Specifically, with respect to tasks $D \rightarrow A$ and $W \rightarrow A$ in Office-31, our proposed strategy separately exceeds the second highest classification accuracy by 2.9% and 1.5%. It demonstrates that collaborative normalization effectively promotes the model generalization ability on target domain. Second, compared with other normalization techniques (BN and TN), CoN successfully scales latent features across various domains to achieve the alignment of different distributions. For example, due to the assistance of CoN, CDAN learns more transferable features which dramatically eliminates domain shift and improves accuracy by 6% and 4.3% on tasks ($Cl \rightarrow Ar$ and $Cl \rightarrow Pr$) when making comparison with BN. Finally, network backbones associated with CoN become more robust for several challenging situations where there exists huge discrepancy about the number of sample within source and target domains. Although *DSLR* or *Webcam* domain has less samples than *Amazon* domain in Office-31, the proposed method still obtains comparable performance, which verifies that the application of collaborative transfer in mini-batch tends to capture more cross-domain information and learn better domain-invariant features. Moreover, CoN layers are also plugged into DANN architecture, which still achieves promising performances in Figure 3. That means it is simple yet effective to deploy our proposed normalization strategy to any frameworks used for domain adaptation.

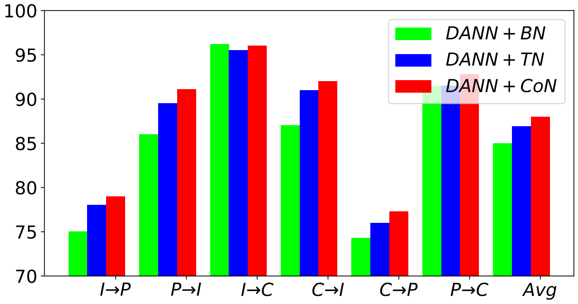


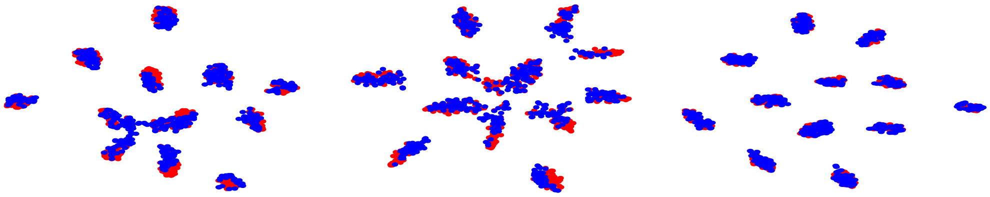
Figure 3: Evaluations of DANN with multiple normalization strategies (BN, TN and CoN).

4.2 Empirical Analysis

Convergence Speed: According to the aforementioned working mechanism, the proposed collaborative normalization adaptively adjusts the direction of feature representations within a mini-batch. Such operation explores cross-domain knowledge to gradually eliminate the discrepancy between source and target domains. Importantly, our strategy effectively accelerates the convergence speed. To clearly illustrate this point, the iterative procedures of CDAN with CoN on tasks $P \rightarrow I$ and $C \rightarrow P$ are reported in Figure 5 (a). We easily find that compared to BN and TransNorm, our method with CDAN rapidly achieves the optimal solution. Concretely, take task $P \rightarrow I$ as an example, collaborative transfer strategy only costs 2,000 iterations to reach the highest classification accuracy,

Table 3: Classification Accuracy (%) on Office-Home dataset (ResNet-50). The best results among all methods are shown with **underline** while the highest accuracy of three normalization tools is in **Red** type.

Method	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**
Res-Net	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	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
DSR	53.4	71.6	77.4	57.1	66.8	69.3	56.7	49.2	75.7	68.0	54.0	79.5	64.9
SymNets	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
TADA	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SAFN	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
CDAN+BN	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+TN	50.2	71.4	77.4	59.3	72.7	73.1	61.0	53.1	79.5	71.9	59.0	82.9	67.6
CDAN+CoN	51.2	72.2	77.5	63.6	74.3	72.0	61.9	56.5	79.8	75.4	55.6	84.2	68.7



(a) CDAN+BN (b) CDAN+TN (c) CDAN+CoN

Figure 4: Visualization of features on task $C \to P$ by using CDAN with multi-normalization tools.

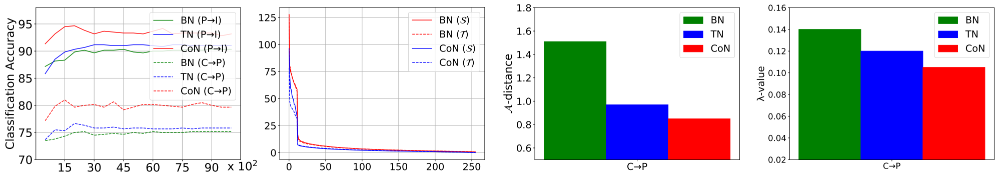
while TN and BN require 3,500 and 4,500 iterations, respectively. The main reason results from the application of cross-domain knowledge which guides features to quickly shrink to the correct direction and makes features more discriminative.

Feature Visualization & Singular Values: To explicitly understand the situation of distribution in abstract semantic space, the t-SNE technique is exploited to visualize feature representations in 2D-panel. The comparative experiments among BN, TN and CoN with CDAN are performed on task $C \to P$ of Image-CLEF. Different from BN and TN, there exists tangible boundary among various categories generated by CoN (Figure 4 (c)). And it is difficult to distinguish source samples from target instances. These experimental performances demonstrate that replacing traditional BN with CoN effectively scales feature representations and dramatically mitigates the influence of domain shift. Moreover, we also explore **SVD** tool to obtain singular values from the learned features on task $C \to P$. As shown in Figure 5 (b), CDAN+CoN has smaller difference between the largest and the smallest singular values compared with CDAN+BN. According to the theory of BSP (Chen et al. (2019), we achieve the conclusion that CoN can learn more **discriminative** features, which is consistent with our theoretical analysis in Sec. 3.3.

Generalization Analysis: Under the adversarial scheme, domain discrepancy is approximated by $\mathcal{A}$-divergence (Ben-David et al. (2010)) with the formulation as:

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{S},\mathcal{T}) = 2\Big(1 - \frac{1}{n_s+n_t}\big(\sum\nolimits_{x:D(x)=0} I(x \in \mathcal{D}_s) + \sum\nolimits_{x:D(x)=1} I(x \in \mathcal{D}_t)\big)\Big), \quad (9)$$

where $D(\cdot)$ means the domain classifier distinguishing source domain from target domain ($D(x \in \mathcal{D}_s) = 1$ and $D(x \in \mathcal{D}_t) = 0$). This indicator reflects the alignment of distribution in the latent space. In addition, in terms of the learning bound theory in (Ben-David et al. (2010)) ($\epsilon_{\mathcal{T}}(h) \leq$



(a) Convergence Speed. (b) Singular Values. (c) $\mathcal{A}$-distance. (d) λ-value.

Figure 5: Visualization of convergence speed, eigen-values, $\mathcal{A}$-distance and λ-value.

$\epsilon_{\mathcal{S}}(h) + \frac{1}{2}d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{S},\mathcal{T}) + \lambda)$, the expected error of hypothesis space h on target domain $\epsilon_{\mathcal{T}}(h)$ is also determined by the error λ of the ideal joint hypothesis h^* on two domains. We apply domain classifier in CDAN with various normalization tools on task $C \rightarrow P$ to evaluate $\mathcal{A}$-divergence and λ in Figure 5 (c) and (d), where CoN with CDAN obtains lower values than BN and TN. It indicates that scaling features with CoN easily learns domain-invariant features and achieve distribution alignment.

5 Conclusion

Unsupervised domain adaptation (UDA) aims to learn model with high generalization ability by achieving domain adaptation. In this paper, we rethink UDA from manifold distribution perspective and propose a novel collaborative normalization strategy suitable for the forward propagation of features to achieve domain alignment. Theoretical and experimental studies fully illustrate that the application of CoN in convolutional layers effectively improves classification performance and accelerates model training convergence on solving UDA issue.

References

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

Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, pp. 2060. NIH Public Access, 2019.

Fabio Maria Cariucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic domain alignment layers. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5077–5085. IEEE, 2017.

Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7354–7362, 2019.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pp. 1081–1090, 2019.

Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*, 2016.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pp. 2960–2967, 2013.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *Domain Adaptation in Computer Vision Applications*, pp. 189–209. Springer, 2017.

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

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pp. 4013–4022, 2019.

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

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

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

You-Wei Luo, Chuan-Xian Ren, Pengfei Ge, Ke-Kun Huang, and Yu-Feng Yu. Unsupervised domain adaptation via discriminative manifold embedding and alignment. *arXiv preprint arXiv:2002.08675*, 2020.

Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8266–8276, 2019.

Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems*, pp. 2558–2567, 2018.

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

Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9471–9480, 2019.

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

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8050–8058, 2019.

Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *AAAI*, pp. 5940–5947, 2020.

Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. *arXiv preprint arXiv:2003.08607*, 2020.

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

Guangrun Wang, Ping Luo, Xinjiang Wang, Liang Lin, et al. Kalman normalization: Normalizing internal representations across network layers. In *Advances in neural information processing systems*, pp. 21–31, 2018.

Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 1951–1961, 2019a.

Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019b.

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1426–1435, 2019.

Weichen Zhang, Dong Xu, Wanli Ouyang, and Wen Li. Self-paced collaborative and adversarial network for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019a.

Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5031–5040, 2019b.