MITIGATING ACCUMULATED DISTRIBUTION DIVERGENCE IN BATCH NORMALIZATION FOR UNSUPERVISED DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Batch Normalization (BN) is a widely used technique in modern deep neural networks that has been proven to be effective in tasks such as Unsupervised Domain Adaptation (UDA) in cross-domain scenarios. However, existing BN variants tend to aggregate source and target domain knowledge in the same channel, which can lead to suboptimal transferability due to unaligned features between domains. To address this issue, we propose a new normalization method called Refined Batch Normalization (RBN), which leverages estimated shift to quantify the difference between estimated population statistics and expected statistics. Our key finding is that the estimated shift can accumulate due to BN stacking in the network, which can adversely affect target domain performance. We further demonstrate that RBN can prevent the accumulation of estimated shift and improve overall performance. To implement this technique, we introduce the RBNBlock, which replaces a BN with RBN in the bottleneck block of a residual network. Our comprehensive experiments on cross-domain benchmarks confirm the effectiveness of RBN in improving transferability across domains.

1 INTRODUCTION

The primary objective of Unsupervised Domain Adaptation (UDA)Singh (2021); Mahapatra et al. (2022); Murez et al. (2018) is the transference of knowledge derived from a labeled source domain to an unlabeled target domain. UDA has found extensive application in diverse domains such as classification, detectionXu et al. (2020), and segmentation Zhou et al. (2020). In pursuit of enhancing feature transferability and acquiring domain-specific insights, scholars have predominantly concentrated their efforts on augmenting the feature normalization module within deep neural networks (DNNs), in conjunction with conventional strategies for feature alignment and pixel-level image translation.

Batch Normalization (BN)Ioffe & Szegedy (2015b) has proven to be a potent remedy for addressing the issue of internal covariate shift in deep neural networks. Nevertheless, recent investigations have illuminated its potential shortcomings in UDA scenarios, chiefly attributed to the shared mean and variance parameters between domains, which is deemed unsuitable. In response to this challenge, several methodologies have been introduced with the aim of preserving domain-specific information. For instance, AdaBNLi et al. (2016c) advocates for the use of distinct domain-specific statistics, albeit a reliance solely on target statistics during inference can result in the forfeiture of source domain insights. AutoDIAL Maria Carlucci et al. (2017b), conversely, mitigates this concern by amalgamating domain statistics on a channel-wise basis through the utilization of shared weight parameters for each channel. On a contrasting note, InterBN Wang et al. (2021) leverages scaling factors originating from individual channels within the BN framework, orchestrating a self-regulating process to determine channel importance, thereby safeguarding domain-specific information.

The previously mentioned approaches have indeed made strides in enhancing Unsupervised Domain Adaptation (UDA) by harnessing domain-specific insights from pertinent channels within the Batch Normalization (BN) framework. However, they still grapple with challenges when confronted with intricate scenarios. A well-documented limitation of BN pertains to its sensitivity to batch size variations, wherein the error rate of BN experiences a rapid escalation as the batch size diminishes Huang

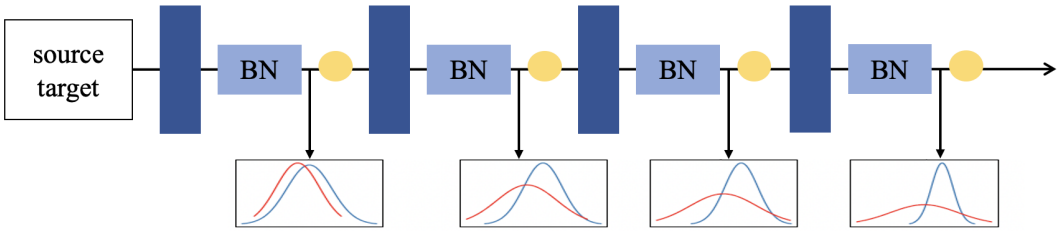


Figure 1: The figure illustrates the main observations, with the orange rectangle and green circle representing linear and non-linear transformations, respectively. By inputting both the source and target domains into the network, it can be observed that as the number of network layers increases, the differences between the two domains become more pronounced.

et al. (2022). Surprisingly, this issue has remained largely unaddressed within the domain adaptation literature.

In this investigation, we draw inspiration from the work of BFN Huang et al. (2022) and introduce an innovative concept: the expected population statistics of BN tailored for UDA. This novel approach takes into consideration the precarious nature of activation population statistics, characterized by a fluctuating distribution during training. When the estimated population statistics of BN deviate from their expected values, we identify this discrepancy as an "estimation shift" of BN.

Our primary discovery underscores that this estimation shift of BN has the potential to accumulate within a neural network, as visually depicted in Figure $\mathbb{I}$. This revelation sheds light on why a network employing BN encounters substantial performance degradation under traditional batch normalization. Moreover, it elucidates the imperative need for adapting the population statistics of BN in response to distribution shifts in the input data during testing. Further investigations reveal that the utilization of BFN Huang et al. (2022), which entails normalizing each sample independently without factoring in the batch dimension, can effectively thwart the accumulation of estimation shift in the context of UDA. Consequently, this mitigates the performance deterioration observed in networks when distribution shifts transpire.

Building upon these insights, we introduce the RBNBlock, a novel architectural component that replaces one instance of BN with Randomized Batch Normalization (RBN) in the bottleneck region of the network. Our contributions in this endeavor can be succinctly summarized as follows:

- Our investigation introduces a novel approach known as Refined Batch Normalization (RBN) tailored specifically for Unsupervised Domain Adaptation (UDA). RBN offers the distinct advantage of augmenting performance without necessitating explicit architectural alterations. An eminent merit of RBN lies in its innate simplicity, as it obviates the requirement for supplementary network modules beyond the core backbone network, rendering its implementation straightforward and computationally efficient.
- RBN stands out as a highly adaptable and efficient technique, seamlessly integrable into a diverse array of UDA methodologies. Its incorporation into these frameworks yields tangible enhancements in their overall performance.
- Rigorous evaluation conducted across multiple cross-domain benchmark datasets, encompassing Office-31, ImageCLED-DA, Office-Home, and VisDA-2017, has consistently underscored the efficacy of RBN. It emerges as a dependable enhancer, consistently delivering notable improvements in performance across these varied evaluation scenarios.

2 Related Work

2.1 Unsupervised Domain Adaptation (UDA)

From a technical standpoint, the formulation of the loss function in the context of Unsupervised Domain Adaptation (UDA) typically involves two predominant approaches.

The first approach revolves around the minimization of cross-domain distribution discrepancies by aligning all statistical aspects between the two domains. Exemplifying this approach, methods such as Deep Domain Confusion (DDC) Tzeng et al. (2014) and Domain Adaptation Network (DAN) Long et al. (2015a) employ the Maximum Mean Discrepancy (MMD) Gretton et al. (2006) as a metric to quantify and mitigate the disjunction between the source and target domains. Concurrently, Joint Maximum Mean Discrepancy (JAN) Long et al. (2017a) innovatively combines adversarial learning principles with MMD. Additionally, methodologies like Sliced Wasserstein Distance (SWD) Lee et al. (2019) and Contrastive Domain Discrepancy (CAN) Kang et al. (2019) introduce Sliced Wasserstein Distance and Contrastive Domain Discrepancy, respectively, as alternative metrics for effectively gauging domain discrepancies.

The second approach entails the incorporation of domain discriminators and harnesses adversarial learning to incentivize domain confusion. Notably, Domain-Adversarial Neural Network (DANN) Ganin & Lempitsky (2015) introduces a domain adversarial loss function to facilitate the acquisition of domain-invariant representations. Adversarial Discriminative Domain Adaptation (ADDA) Tzeng et al. (2017a) further integrates adversarial learning with discriminative feature learning by employing asymmetric feature extractors tailored to each domain. Conditional Domain Adversarial Network (CDAN) Long et al. (2018a) adopts a conditional domain-adversarial paradigm in the training of adversarial adaptation models.

Notwithstanding the breadth of existing methodologies, a common limitation pervades them all: the oversight of misalignment issues among corresponding channels across domains. This misalignment concern frequently gives rise to suboptimal outcomes in domain adaptation performance.

2.2 Normalization Techniques

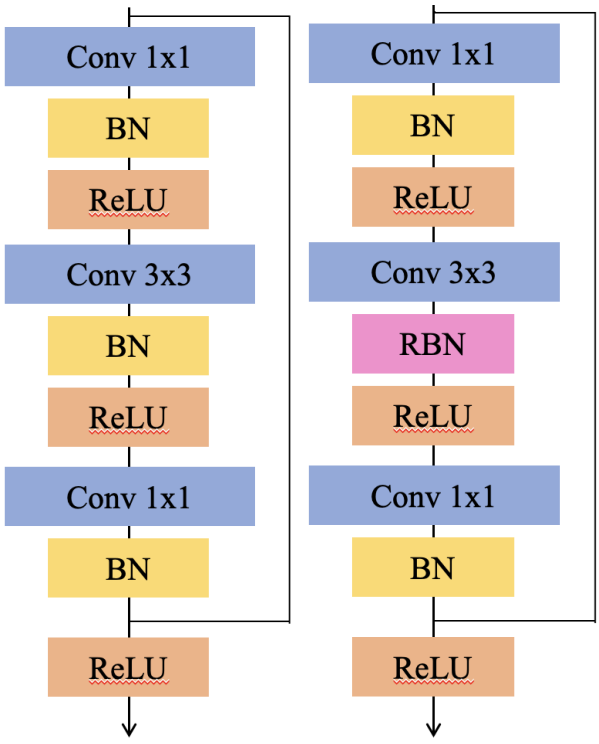
Normalization techniques serve as integral components within Convolutional Neural Networks (CNNs), playing a pivotal role in augmenting their learning efficiency, stability, and generalization capabilities Tseng et al. (2020); Wang et al. (2019a); Du et al. (2020). Among the array of widely adopted methods, one finds Batch Normalization (BN) Ioffe & Szegedy (2015a), Layer Normalization (LN) Ba et al. (2016a), Adaptive Batch Normalization (AdaBN) Li et al. (2016a), Group Normalization (GN) Wu & He (2018), Switchable Normalization (SN) Luo et al. (2018), TaskNorm Bronskill et al. (2020), EvoNorm Liu et al. (2020), Meta-Norm Du et al. (2020), and Representation Normalization Gao et al. (2021).

Within the domain adaptation paradigm, researchers have devised novel normalization strategies to improve adaptation performance. These innovations encompass the likes of AdaBN Li et al. (2016a), which leverages source and target domain statistics separately, AutoDIAL Maria Carlucci et al. (2017a), facilitating statistics integration on a channel-by-channel basis, DSBN Chang et al. (2019), which normalizes source and target representations individually, TN Wang et al. (2019a), employing channel attention mechanisms to align source and target statistics, ConvNorm Li & Vasconcelos (2019), conducting domain adaptation via a specialized adaptation layer, and DWT Roy et al. (2019), which employs dual co-variance matrices to whiten feature maps from both domains individually.

However, it is important to note that these prevailing methodologies exhibit a common limitation - they disregard the issue of misalignment among non-corresponding channels across domains. In this study, we present an innovative feature normalization method that confronts this misalignment concern through the enactment of cross-domain reciprocity. Diverging from conventional normalization techniques, which either segregate feature maps or solely account for corresponding cross-domain channels, our approach delves into modeling the non-corresponding channels as a key facet of domain adaptation.

3 Method

In the context of Unsupervised Domain Adaptation (UDA), the formal representation of the source and target domains is established as follows: We denote the source domain as $\mathbf{S} = \{(x_{sk}, y_{sk})\}_{k=1}^{m_s}$, which comprises a collection of m_s labeled samples, where the labels $y_{sk} \in \{1, 2, \ldots, N\}$ correspond to the source domain data instances x_{sk}. Simultaneously, the target domain is denoted as $\mathbf{T} = \{x_{ti}\}_{i=1}^{m_t}$, encompassing m_t unlabeled samples. It is imperative to underscore that the labels



(a) Original bottleneck (b) RBNBlock

Figure 2: We present the key findings through a visual representation. A linear and non-linear transformation are depicted by an orange rectangle and green round, respectively. We demonstrate the distribution of normalized output at each BN layer for both the source and target data with distribution shift, and measure the degree of difference between the estimated population statistics and the expected values.

assigned to the source domain samples are defined over the set $\{1, 2, \ldots, N\}$, and it is essential to acknowledge that the source and target data are sampled from distinct probability distributions.

3.1 Batch Normalization

Within the framework of a multi-layer perceptron (MLP), we denote the p-dimensional input to a specific layer as $x \in \mathbb{R}^p$. During the training phase, the utilization of batch normalization [6] entails the normalization of each neuron or channel by considering a mini-batch of size n, and this process is outlined as follows:

$$\hat{x}_i = BN\left(x_i\right) = \frac{x_i - \nu_i}{\sqrt{\tau_i^2 + \epsilon}}, i = 1, 2, \ldots, p \tag{1}$$

Batch normalization involves the computation of mini-batch statistics, namely the mini-batch mean denoted as ν_i and the variance represented as τ_i^2, for each neuron. Specifically, ν_i corresponds to the average value of n samples of x_i, while τ_i^2 is the variance of n samples of $\left(x_i - \nu_i\right)^2$. To circumvent potential issues of numerical instability, a small constant ϵ is added to both the mean and variance calculations.

However, during the inference or testing phase, there arises a need to ascertain the population statistics, namely the population mean denoted as $\tilde{\nu}$ and the population variance represented as $\tilde{\tau}^2$, of the layer input. This determination of population statistics is essential for batch normalization to make deterministic predictions.

$$\hat{x}_i = BN_{\text{inf}}\left(x_i\right) = \frac{x_i - \bar{\nu}_i}{\sqrt{\tau_i^2}}, i = 1, 2, \ldots, p \tag{2}$$

While the direct computation of population statistics $\left\{\tilde{\nu}, \tilde{\tau}^2\right\}$ for the layer input is not feasible, we can approximate these statistics using $\left\{\hat{\nu}, \hat{\tau}^2\right\}$, which represent the running averages of mini-batch statistics accumulated over various training iterations denoted by s. To update and maintain these estimates, we employ an update factor denoted as α, and the calculation proceeds as follows:

$$\begin{cases} \hat{\nu}^s = (1-\alpha)\hat{\mu}^{s-1} + \alpha\nu^{s-1} \\ \left(\hat{\tau}^s\right)^2 = (1-\alpha)\left(\hat{\tau}^{s-1}\right)^2 + \alpha\left(\tau^{s-1}\right)^2. \end{cases} \tag{3}$$

The discordance in Batch Normalization (BN) behavior between training and inference poses a challenge, particularly in recurrent neural networks, and it detrimentally affects performance when dealing with small batch sizes, mainly due to the potential inaccuracies in population statistics estimation, as highlighted in prior work Huang et al. (2022).

To circumvent the necessity of estimating population statistics, a novel method known as Batch-Free Normalization (BFN) Huang et al. (2022) has been devised. BFN operates by refraining from normalization along the batch dimension, thereby ensuring consistent operations during both training and inference phases. An illustrative example of such an approach is Layer Normalization (LN) Ba et al. (2016b), which standardizes the layer input within the neurons for each training sample by:

$$\hat{x}_i = LN\left(x_i\right) = \frac{x_i - \nu}{\sqrt{\tau^2 + \epsilon}}, \quad i = 1, 2, \ldots, p \tag{4}$$

In this context, the symbol $\nu = \frac{1}{p}\sum_{i=1}^{p} x_i$ represents the mean, and $\tau^2 = \frac{1}{p}\sum_{i=1}^{p}\left(x_i - \nu\right)^2$ denotes the variance of each sample. Layer Normalization (LN) [35] extends this concept by standardizing the layer input across neurons while independently normalizing neurons within designated groups. In contrast, Group Normalization (GN) offers greater flexibility than LN by permitting the adjustment of the number of groups. GN has proven effective, particularly in visual tasks with small batch training.

Recent research by Huang et al. Huang et al. (2022) has introduced Batch-Free Normalization (BFN), a method that prevents the accumulation of estimation drift. BFN independently normalizes each sample instead of performing normalization across the batch dimension, thereby mitigating the deterioration of network performance in the presence of distributional drift.

Inspired by the principles underlying BFN, we approach the Unsupervised Domain Adaptation (UDA) problem from the perspective of refining batch normalization techniques.

3.2 Refine Batch Normalization (RBN)

In recent years, several methods have been proposed to address the limitations of BN, including AdaBN Li et al. (2016a), AutoDIAL Maria Carlucci et al. (2017a), DWT Roy et al. (2019), DSBN Chang et al. (2019), and TN Wang et al. (2019a). Figure \illustrates the main differences between these typical UDA normalization techniques and our RBN. Generally, separate normalization is adopted by these methods to avoid sharing the exact same mean and variance. However, this mechanism suffers from estimation bias, where the estimated population statistics of BN do not match the expected statistics. Investigating the impact of estimation bias on the performance of batch normalization networks is of significant importance. Therefore, this paper attempts to quantitatively measure the degree of difference between the estimated and expected population statistics in UDA.

We define $\tilde{\nu}\left(\tilde{\tau}^2\right)$ as the expected population mean (variance) of BN and $\hat{\nu}\left(\hat{\tau}^2\right)$ as the estimated value. The magnitude of estimation shift is quantified by the L^2-norm of their difference, $ESM_\nu = \|\hat{\nu} - \tilde{\nu}\|_2$ and $ESM_\tau = \left\|\sqrt{\hat{\tau}^2} - \sqrt{\tilde{\tau}^2}\right\|_2$.

This paper also features an experiment that aims to examine how batch normalization estimation bias affects UDA networks' performance, and presents potential corrective measures.

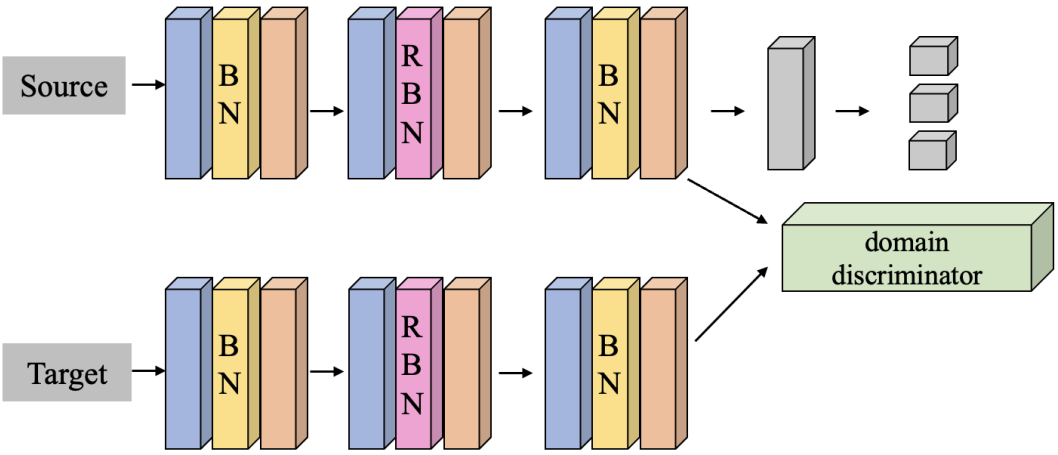
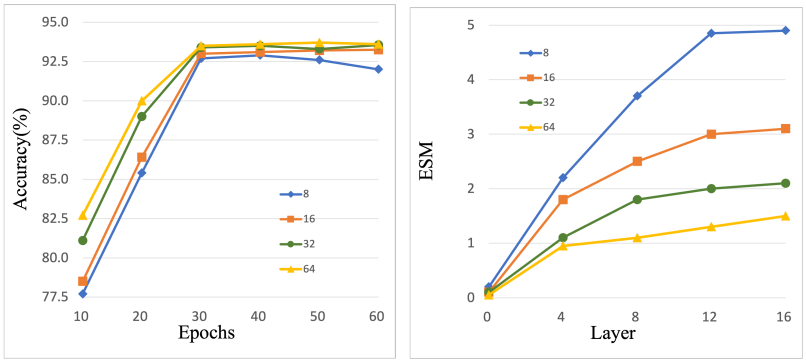


Figure 3: The proposed UDA method employs a framework in which both the source and target domains are inputted into the network. A substitution of the BN layer with the RBN layer is made. ResNet-50 serves as the primary feature representation network, except for VisDA-2017 where ResNet-101 is utilized.



(a) (b)

Figure 4: (a) demonstrates that the larger the batch size, the better the performance of the network. (b) shows that the ESM gradually increases as the number of layers increases.

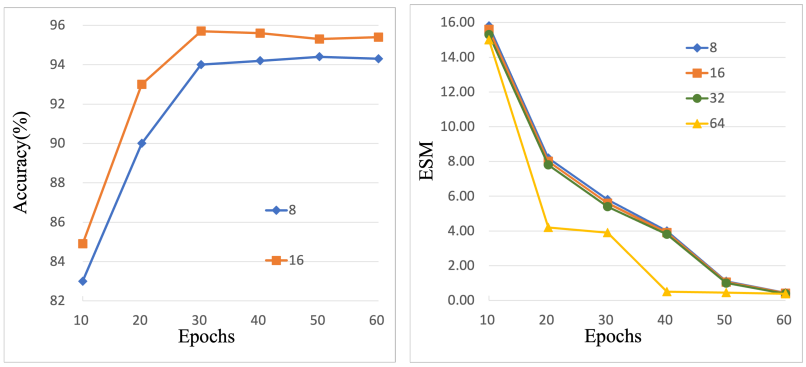
3.3 Evaluation on Visual Recognition Tasks

In this section, we focus on conducting experiments to investigate how the estimation shift of BN impacts the performance of batch normalization networks, and explore potential solutions. BFN findings suggest that the primary reason for the discrepancy in error between training and testing is

the inaccurate estimation of BN's overall statistical data. This estimation bias can accumulate and worsen as the number of BN layers increases. Additionally, BFN indicates that the distribution shift of inputs between the training and testing sets can also result in estimation bias, further negatively impacting testing performance. Finally, we have observed a potential increase in the value of ESM_τ in deeper BN layers at the end of training.

In summary, according to the experiments above, we argue that estimation shift of BN can be potentially accumulated in a network with stacked BNs, which probably has a deteriment effect on the test performance of the network, especially with the distribution shift occurred.

We take CDAN as the baseline, set the batch size to $\{8, 16, 32, 64\}$, and set the epoch to $\{10, 20, 30, 40, 50, 60\}$. As shown in Figure. (a), we can see that the batch size has a certain impact on performance. At the same time, we can see from Figure. (b) that as the number of layers increases, the model shows a stable result. These observations also indicate that the distribution shift



(a) (b)

Figure 5: (a) also demonstrate that the larger the batch size, the better the performance of the network. (b) indicates that by replacing deeper BN layers with GN, ESM can be significantly reduced.

Table 1: Classification accuracy (%) on Office-31 dataset with ResNet-50 as the backbone.

Methods	A → W	D → W	W → D	A → D	D → A	W → A	Avg
Source Only He et al. (2016)	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DDC Tzeng et al. (2014)	75.6	96.0	98.2	76.5	62.2	61.5	78.3
DAN Long et al. (2015b)	80.5	97.1	99.6	78.6	63.6	82.8	80.4
RTN Long et al. (2016)	84.5	96.8	99.4	77.5	66.2	64.8	81.6
DANN Ganin et al. (2016)	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA Tzeng et al. (2017b)	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN Long et al. (2017b)	85.4	97.4	99.8	84.7	68.6	70.0	84.3
MADA Pei et al. (2018)	90.0	97.4	99.6	87.8	70.3	66.4	85.2
MCD Saito et al. (2018)	88.6	98.5	**100.0**	92.2	69.5	69.7	86.5
DWL Xiao & Zhang (2021)	89.2	99.2	**100.0**	91.2	73.1	69.8	87.1
TADA Wang et al. (2019b)	94.3	98.7	99.8	91.6	72.9	73.0	88.4
SHOT Liang et al. (2021)	90.1	98.7	99.9	93.9	75.3	75.0	88.8
SymNet Zhang et al. (2019)	95.2	98.8	**100.0**	93.9	74.6	72.5	88.4
SAR Wang & Zhang (2020)	95.2	98.6	**100.0**	91.7	74.5	73.7	89.0
CDAN Long et al. (2018b)	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
CDAN+RBN	**95.9**	**99.1**	**100.0**	**95.7**	**76.1**	**74.5**	**90.2**

of inputs between the source domain and target domain in domain adaptation can lead to estimation bias of BN, which has a negative impact on domain adaptation performance, and the deeper BN, ESMs have the potential for higher value at the end of training.

We replace deeper BN layers with GN layers, and name the resulting network RBN. RBN can prevent the accumulation of estimation drift in BN in the presence of distribution shift, and mitigate the degradation of network performance. The specific application of RBN on CDAN is shown in the Supplementary Materials.

4 Experiments

4.1 Datasets

Our experiments is performed on four datasets, namely Office-31, Office-Home, ImageCLEF-DA, and VisDA-2017. These datasets are well-known benchmarks that have been extensively used in prior studies to assess domain adaptation algorithms.

Table 2: Classification accuracy (%) on ImageCLEF-DA dataset with ResNet-50 as the backbone.

Methods	I → P	P → I	I → C	C → I	C → P	P → C	Avg
Source Only He et al. (2016)	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN Long et al. (2015b)	74.5	82.2	92.8	86.3	69.2	89.8	82.5
RTN Long et al. (2016)	75.6	86.8	95.3	86.9	72.7	92.2	84.9
DANN Ganin et al. (2016)	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN Long et al. (2017b)	76.8	88.0	94.7	89.5	74.2	91.7	85.8
MADA Pei et al. (2018)	75.0	87.9	96.0	88.8	75.2	92.2	85.8
SAFN Xu et al. (2019)	78.0	91.7	96.2	91.1	77.0	94.7	88.1
SAR Wang & Zhang (2020)	78.3	91.3	96.7	90.5	78.1	96.2	88.5
CDAN+RN Huang et al. (2023)	78.6	92.7	97.2	92.8	79.1	94.8	89.2
CDAN Long et al. (2018b)	77.7	90.7	97.7	91.3	74.2	94.3	87.7
CDAN+RBN	**81.5**	**93.6**	**98.2**	**94.5**	**81.3**	**96.4**	**90.9**

Table 3: Classification accuracy (%) on Office-Home dataset with ResNet-50 as the backbone.

Methods	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg
Source Only He et al. (2016)	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
DAN Long et al. (2015b)	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	57.7	63.1	51.5	74.3	56.3
DANN Ganin et al. (2016)	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
JAN Long et al. (2017b)	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
TADA Wang et al. (2019b)	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
SymNet Zhang et al. (2019)	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
ATM Li et al. (2020)	52.4	72.6	78.0	61.1	72.0	72.6	59.5	52.0	79.1	73.3	58.9	83.4	67.9
CDAN Long et al. (2018b)	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+RBN	**53.8**	**73.5**	**79.4**	**63.2**	**72.9**	**75.7**	**66.3**	**54.2**	**81.3**	**74.5**	**62.9**	**84.8**	**70.2**

Office-31 Saenko et al. (2010) dataset serves as a widely used benchmark for evaluating domain adaptation methods. It includes 4,652 images categorized into 31 classes from three distinct domains: Amazon (A), Dslr (D), and Webcam (W). The images from Amazon are collected from the internet, while those from Webcam and Dslr are manually captured in an office setting. We assess all the compared methods across all six transfer tasks.

ImageCLEF-DA [I] dataset serves as a benchmark for evaluating domain adaptation methods in ImageCLEF 2014. It comprises 12 common categories that are shared by three public datasets: Caltech256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P), with each domain containing 12 categories and 600 images. We assess the methods across all six transfer tasks.

Office-Home Li et al. (2019) dataset comprises four distinct domains, namely $\mathrm{Art}(\mathbf{A})$, Clipart (C), Product (P), and Real World $(\mathbf{R})$, with each domain containing 65 different categories. The dataset contains a total of 15,500 images, making it a large-scale benchmark for evaluating domain adaptation methods. We assess the methods across all twelve transfer tasks.

VisDA-2017 Peng et al. (2017) dataset presents a difficult simulation-to-real scenario, featuring two very distinct domains: synthetic renderings of 3D models captured from varying angles and lighting

Table 4: Classification accuracy (%) on VisDA-2017 dataset with ResNet-101 as the backbone.

Methods	aero	truck	train	skate	person	plant	motor	knife	horse	car	bus	bicycle	Avg
Source Only Long et al. (2018b)	55.1	8.5	73.5	26.5	31.2	81.0	79.7	17.9	80.6	59.1	61.9	53.3	52.4
DAN Long et al. (2015b)	87.1	20.7	85.8	36.3	53.1	49.7	63.0	42.9	90.3	42.0	76.5	63.0	59.2
DANN Ganin et al. (2016)	81.9	7.8	82.8	54.6	65.1	51.9	65.1	29.5	81.2	44.3	82.8	77.7	60.4
MCD Saito et al. (2018)	87.0	25.8	83.0	40.3	76.9	88.6	84.7	79.6	88.9	64.0	83.7	60.9	72.0
BSP+DANN Chen et al. (2019)	92.2	37.1	84.5	66.9	72.4	80.6	86.8	54.0	87.0	47.5	83.8	72.5	72.1
BSP+CDAN Chen et al. (2019)	92.4	38.4	82.1	77.9	77.0	84.2	90.1	80.6	89.0	57.5	81.0	61.0	75.9
DSAN Zhu et al. (2020)	90.9	39.4	89.1	67.6	75.1	92.8	93.7	77.0	88.9	62.4	75.7	66.9	75.1
DWL Xiao & Zhang (2021)	90.1	28.7	85.6	57.1	78.0	90.6	86.8	81.5	92.4	67.6	86.1	80.2	77.1
CDAN Long et al. (2018b)	85.2	38.0	81.9	76.0	74.5	83.4	88.1	74.9	84.2	50.8	83.0	66.9	74.0
CDAN+RBN	**95.9**	**46.7**	**81.3**	**79.8**	**80.1**	**93.7**	**94.8**	**84.2**	**96.7**	**73.8**	**87.7**	**76.3**	**82.6**

Table 5: Accuracy comparison on Office-31 with BN, AdaBN, AutoDIAL and CDAN+RBN.

Methods	A → W	D → W	W → D	A → D	D → A	W → A	Avg
BN Ioffe & Szegedy (2015b)	82.0	96.9	99.1	79.7	68.2	67.4	82.2
AdaBN Li et al. (2016b)	82.4	97.7	99.8	81.0	67.2	68.2	82.7
AutoDIAL Maria Carlucci et al. (2017b)	84.8	97.7	**100.0**	85.7	63.9	68.7	83.5
TransNorm Wang et al. (2019b)	91.8	97.7	**100.0**	88.0	68.2	70.4	86.0
CDAN+RBN	**95.9**	**99.1**	**100.0**	**95.7**	**76.1**	**74.5**	**90.2**

conditions, and real natural images. The dataset includes 12 classes across the training, validation, and test domains.

4.2 Implementation Details

To assess the effectiveness of RBN, we choose CDAN Long et al. (2018b) as our baseline and refer to our proposed model as CDAN+RBN. Our implementation utilizes Pytorch and optimizes the model using minibatch stochastic gradient descent (SGD) with a weight decay of 5×10^{-4} and momentum of 0.9 . The learning rate is set to 10^{-3}. We employ ResNet-50 as the backbone for Office-31, ImageClEF-DA, and Office-Home, and pre-trained ResNet-101 for VisDA-2017 to extract features. We replace the BN layers with GN, utilizing RBNBlocks throughout the model. Specifically, we utilize all labeled source data and all unlabeled target data and report the average classification accuracy across 5 random experiments for each task. All other training settings remain the same.

4.3 Results

The results of our CDAN+RBN approach are presented in Table ??, where we highlight substantial improvements over the baseline CDAN method. Our method yields noteworthy average accuracy enhancements of 2.5%, 3.2%, and 4.4% for the Office-31, ImageCLEF-DA, and Office-Home datasets, respectively. Particularly impressive improvements are observed in some of the most challenging tasks within the Office-31 dataset, such as the transformation from category D ($\mathbf{D}$) to category A ($\mathbf{A}$), where accuracy increases from 71.0% to 76.1%, and from category W ($\mathbf{W}$) to category A ($\mathbf{A}$), where accuracy improves from 69.3% to 74.5%. Furthermore, our InterBN demonstrates remarkable performance on the large-scale VisDA2017 dataset, exhibiting an 8.6% improvement over the baseline CDAN method. Notably, our approach attains superior average classification performance compared to all other state-of-the-art methods across all four benchmark datasets. It is important to highlight that both our method and CDAN share the same fundamental adversarial networks, with the primary distinction being that our approach replaces BN with RBN. Consequently, the observed accuracy enhancements over CDAN can be attributed to the contributions of RBN.

In the context of normalization modules, RBN is engineered to function as an end-to-end trainable layer, fostering improved generalizability. To isolate and showcase the effectiveness of RBN, we conduct a comparative analysis with other normalization methods, including vanilla BN, AdaBN, AutoDIAL, and TransNorm. For a fair comparison, we replace the normalization module with its sibling counterparts while keeping all other network components unchanged.

5 Conclusion

This study makes a significant discovery regarding the accumulation of estimation shift within a network when employing Batch Normalization (BN). This accumulation of estimation shift can have a detrimental impact on a network's performance during testing, especially when distributional shifts occur. Our proposed method is easily integrable into various network architectures by simply substituting the BN layer with the RBN module during the training process. Empirical investigations underscore the substantial enhancement brought about by InterBN when incorporated into existing domain adaptation techniques, resulting in notably improved accuracy across various benchmark datasets.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016a.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. corr abs/1607.06450 (2016). *arXiv preprint arXiv:1607.06450*, 178, 2016b.

John Bronskill, Jonathan Gordon, James Requeima, Sebastian Nowozin, and Richard Turner. Tasknorm: Rethinking batch normalization for meta-learning. In *International Conference on Machine Learning*, pp. 1153–1164. PMLR, 2020.

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/CVF 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. PMLR, 2019.

Yingjun Du, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Metanorm: Learning to normalize few-shot batches across domains. In *International Conference on Learning Representations*, 2020.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Shang-Hua Gao, Qi Han, Duo Li, Ming-Ming Cheng, and Pai Peng. Representative batch normalization with feature calibration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8669–8679, 2021.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

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.

Lei Huang, Yi Zhou, Tian Wang, Jie Luo, and Xianglong Liu. Delving into the estimation shift of batch normalization in a network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 763–772, 2022.

Zhiyong Huang, Kekai Sheng, Ke Li, Jian Liang, Taiping Yao, Weiming Dong, Dengwen Zhou, and Xing Sun. Reciprocal normalization for domain adaptation. *Pattern Recognition*, 140:109533, 2023.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015a.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015b.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4893–4902, 2019.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10285–10295, 2019.

Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Zi Huang. Cycle-consistent conditional adversarial transfer networks. In *Proceedings of the 27th ACM international conference on multimedia*, pp. 747–755, 2019.

Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. Maximum density divergence for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3918–3930, 2020.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016a.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016b.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016c.

Yunsheng Li and Nuno Vasconcelos. Efficient multi-domain learning by covariance normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5424–5433, 2019.

Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021.

Hanxiao Liu, Andy Brock, Karen Simonyan, and Quoc Le. Evolving normalization-activation layers. *Advances in Neural Information Processing Systems*, 33:13539–13550, 2020.

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

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

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016.

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

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

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018a.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018b.

Ping Luo, Jiamin Ren, Zhanglin Peng, Ruimao Zhang, and Jingyu Li. Differentiable learning-to-normalize via switchable normalization. *arXiv preprint arXiv:1806.10779*, 2018.

Dwarikanath Mahapatra, Steven Korevaar, Behzad Bozorgtabar, and Ruwan Tennakoon. Unsupervised domain adaptation using feature disentanglement and gcns for medical image classification. In *European Conference on Computer Vision*, pp. 735–748. Springer, 2022.

Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *Proceedings of the IEEE international conference on computer vision*, pp. 5067–5075, 2017a.

Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *Proceedings of the IEEE international conference on computer vision*, pp. 5067–5075, 2017b.

Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4500–4509, 2018.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

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/CVF 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 *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pp. 213–226. Springer, 2010.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:5089–5101, 2021.

Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv:2001.08735*, 2020.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017a.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017b.

Mengzhu Wang, Wei Wang, Baopu Li, Xiang Zhang, Long Lan, Huibin Tan, Tianyi Liang, Wei Yu, and Zhigang Luo. Interbn: Channel fusion for adversarial unsupervised domain adaptation. In *Proceedings of the 29th ACM international conference on multimedia*, pp. 3691–3700, 2021.

Shanshan Wang and Lei Zhang. Self-adaptive re-weighted adversarial domain adaptation. *arXiv preprint arXiv:2006.00223*, 2020.

Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. *Advances in neural information processing systems*, 32, 2019a.

Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. *Advances in neural information processing systems*, 32, 2019b.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Ni Xiao and Lei Zhang. Dynamic weighted learning for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15242–15251, 2021.

Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12355–12364, 2020.

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/CVF international conference on computer vision*, pp. 1426–1435, 2019.

Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5031–5040, 2019.

Wei Zhou, Yukang Wang, Jiajia Chu, Jiehua Yang, Xiang Bai, and Yongchao Xu. Affinity space adaptation for semantic segmentation across domains. *IEEE Transactions on Image Processing*, 30:2549–2561, 2020.

Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems*, 32(4):1713–1722, 2020.