ESBN: Estimation Shift of Batch Normalization for Source-free Universal Domain Adaptation

Jiao Li[1†], **Houcheng Suu**[2†], **Bingli Wang**[3], **Yuandong Min**[4], **Mengzhu Wang**[5*], **Nan Yin**[5], **Shanshan Wang**[6] and **Jingcai Guo**[7*]
[1]University of Electronic Science and Technology of China
[2]Hong Kong University of Science and Technology (Guangzhou)
[3]Sichuan Agricultural University
[4]Central China Normal University
[5]Hebei University of Technology
[6]Anhui University
[7]The Hong Kong Polytechnic University
jiaoli.research@outlook.com, dreamkily@gmail.com, jingcai.guo@gmail.com

Abstract

Domain adaptation (DA) is crucial for transferring models trained in one domain to perform well in a different, often unseen domain. Traditional methods, including unsupervised domain adaptation (UDA) and source-free domain adaptation (SFDA), have made significant progress. However, most existing DA methods rely heavily on Batch Normalization (BN) layers, which are not optimal in source-free settings, where the source domain is unavailable for comparison. In this study, we propose a novel method, ESBN, which addresses the challenge of domain shift by adjusting the placement of normalization layers and replacing BN with Batch-free Normalization (BFN). Unlike BN, BFN is less dependent on batch statistics and provides more robust feature representations through instance-specific statistics. We systematically investigate the effects of different BN layer placements across various network configurations and demonstrate that selective replacement with BFN improves generalization performance. Extensive experiments on multiple domain adaptation benchmarks show that our approach outperforms state-of-the-art methods, particularly in challenging scenarios such as Open-Partial Domain Adaptation (OPDA).

1 Introduction

Deep learning exhibits excellent performance when the test domain matches the training set; however, performance deteriorates when a domain shift occurs. Domain adaptation (DA) allows the model to maintain its performance in the target domain by mitigating the domain shift. DA has evolved through several paradigms, each addressing increasingly complex scenarios. Classical unsupervised domain adaptation (UDA) methods initially focused on situations with shared label spaces between domains, employing various techniques such as adversarial training [Ganin *et al.*, 2016; Rangwani *et al.*, 2022] and feature alignment [Tranheden *et al.*, 2021; **?**; Wang *et al.*, 2024c] to bridge the domain gap. However, these approaches often face a fundamental limitation: the assumption of identical label spaces between source and target domains rarely holds in real-world scenarios [Liang *et al.*, 2020; Qu *et al.*, 2022; Qu *et al.*, 2024a].

To address this limitation, researchers have proposed several specialized frameworks. Partial Domain Adaptation (PDA) [Cao *et al.*, 2019] handles cases where target labels form a subset of source labels, while Open-set Domain Adaptation (OSDA) [Liu *et al.*, 2019] deals with scenarios where the target domain contains unknown categories. Open-Partial Domain Adaptation (OPDA) [You *et al.*, 2019; Saito and Saenko, 2021] combines both challenges, where the target domain contains unknown categories and only a partial set of labels from the source domain. However, these methods still require prior knowledge about the relationship between source and target label spaces in Universal Domain Adaptation (UniDA) [Li *et al.*, 2024]. UniDA [You *et al.*, 2019; Fu *et al.*, 2020] represents a significant advancement by eliminating the need for such prior knowledge. It tackles the most general case where label spaces may overlap arbitrarily between domains, without assuming any information about the nature or extent of this overlap. This flexibility makes UniDA particularly relevant for real-world applications where label relationships between domains are unknown or uncertain. Recent developments have further considered practical constraints, particularly data privacy concerns [Wang *et al.*, 2024d]. Source-free Universal Domain Adaptation (SF-UniDA) [Liu and Zhou, 2024; Qu *et al.*, 2024b; Schlachter and Yang, 2024] addresses scenarios where direct access to source domain data is restricted, working instead with a pre-trained source model.

In the current study, some researchers have attempted to

*Corresponding author.

improve Batch Normalization (BN) for addressing domain shift, it is important to note that a model's feature extraction capability remains fundamentally unchanged across domains. The performance degradation primarily arises from the invalidation of previously calculated BN statistics when encountering domain shifts, which subsequently affects the quality of feature representation. This observation has prompted various approaches that focus on adjusting BN statistics to mitigate the effects of domain shift [Liu *et al.*, 2021; Li *et al.*, 2024]. For instance, methods like batch normalization statistics transfer explicitly align BN statistics between source and target domains [Fang *et al.*, 2024]. Other researchers have addressed domain adaptation by updating only the affine parameters in the BN layer, while preserving the source domain statistics to reduce label-domain interference and enhance domain knowledge learning [Wu *et al.*, 2024].

Although numerous approaches have been proposed to address unsupervised domain adaptation by adjusting BN, relatively fewer studies focus on source-free domain adaptation. Existing research predominantly concentrates on altering the behavior of BN, which is not directly applicable to source-free unsupervised domain adaptation. This is because, unlike unsupervised domain adaptation, source-free unsupervised domain adaptation lacks a source domain. Many unsupervised domain adaptation methods rely on comparing the BN statistics between the source and target domains for domain adaptation. However, in the absence of a source domain, it is challenging to directly compare the statistics between the source and target domains. Directly modifying BN is difficult to apply in source-free domain adaptation, and BN layers, which overly rely on batch statistics, are ineffective in reducing domain shift in source-free domain adaptation tasks. Additionally, we observe that the placement of BN layers significantly impacts the overall generalization ability of the model, as different positions correspond to different feature representations. Existing research primarily focuses on altering the behavior of BN, overlooking the role of BN layer placement and its impact on cross-domain generalization.

To address these issues and the limitations of source-free domain adaptation, we propose ESBN. By addressing the model's feature representation impact caused by BN statistics shift due to domain offset, we aim to reduce the influence of domain shift on BN layers. Specifically, we explore adjusting the position of the normalization layers and replacing them with a more robust, batch-independent normalization method to mitigate the impact of domain shift on BN layers, resulting in a more robust model. We introduce Batch-free Normalization (BFN), which, compared to batch-dependent statistics, can potentially provide more robust feature representations through instance-specific statistics. We systematically analyze the effectiveness of replacing BN with BFN at various network positions (such as odd layers, even layers, and all layers) and configurations (such as alternating modes). The goal of this strategy is to achieve an optimal balance between maintaining discriminative power and reducing domain-specific biases during feature learning. Furthermore, we investigate the significant impact of the placement of normalization layers on cross-domain feature learning and identify a more suitable placement for these layers. Overall, our approach demonstrates superior performance compared to the state-of-the-art across multiple experiments. The main contributions can be summarized as follows:

- We present the first comprehensive study on the impact of normalization layer placement in domain adaptation, revealing how different configurations affect feature transferability and domain generalization.
- We propose an approach for integrating Batch-free Normalization into existing architectures, demonstrating that selective replacement of BN layers can significantly improve generalization performance.
- We conduct extensive experiments across multiple domain adaptation benchmarks, providing empirical evidence that our strategic normalization approach outperforms traditional DA methods, including in challenging OPDA scenarios.

2 Related Work

Unsupervised Domain Adaptation. Unsupervised Domain Adaptation (UDA) focuses on aligning source and target domains by learning domain-invariant feature representations [Peng *et al.*, 2019; Zellinger *et al.*, 2017; Wang *et al.*, 2024a; Wang *et al.*, 2024b]. This challenge is tackled by utilizing labeled data from source domains to train models for unlabeled target domains in a transductive learning framework. Existing UDA methods can be categorized into two primary categories: *(1) Moment matching approaches*, such as Maximum Mean Discrepancy (MMD) and the Wasserstein metric, focus on aligning statistical properties between source and target distributions [Tranheden *et al.*, 2021; **?**]. *(2) Adversarial learning approaches* employ domain discriminators to promote domain-invariant feature extraction [Saito *et al.*, 2018; Sankaranarayanan *et al.*, 2018; Zhang *et al.*, 2019; Xiao *et al.*, 2021; Rangwani *et al.*, 2022]. However, the majority of existing methods operate under the assumption that the label spaces of the source and target domains are identical. And despite the extensive theoretical advancements in Unsupervised Domain Adaptation (UDA), the acquisition of substantial source data remains impractical. This assumption restricts their applicability in real-world scenarios.

Source-Free domain adaptation. Source-Free Domain Adaptation (SFDA) is a domain adaptation approach aimed at leveraging pre-trained source models to facilitate knowledge transfer to the target domain in the absence of source data [Tzeng *et al.*, 2017; Hoffman *et al.*, 2018]. Existing SFDA methods can be categorized into two classes. *Data generation techniques* that synthesize source data [Hou and Zheng, 2020; Qiu *et al.*, 2021; Du *et al.*, 2024] and *model fine-tuning approaches* that utilize unlabeled target data within a self-supervised framework [Liang *et al.*, 2020; Yang *et al.*, 2021b; Yang *et al.*, 2021a]. However, many current SFDA methods are limited to closed-set scenarios, which constrains their practical applicability. Although some studies have explored open-partial-set scenarios, they remain insufficient [Liang *et al.*, 2021].

3 Methodology

3.1 Preliminary

Problem Definition. We address universal domain adaptation (UniDA), which tackles both covariate and label distribution shifts between domains. Given a source domain $\mathcal{S} = \{(\mathbf{v}_i^s, \mathbf{z}_i^s)\}_{i=1}^{M_s}$ with $\mathbf{v}_i^s \in \mathcal{A}^s \subset \mathbb{R}^d$ and $\mathbf{z}_i^s \in \mathcal{Z}^s \subset \mathbb{R}^K$, and an unlabeled target domain $\mathcal{T} = \{(\mathbf{v}_i^t)\}_{i=1}^{M_t}$ with $\mathbf{v}_i^t \in \mathcal{A}^t \subset \mathbb{R}^d$. Let $\mathcal{Z} = \mathcal{Z}^s \cap \mathcal{Z}^t$ denote shared labels, while $\hat{\mathcal{Z}}^s = \mathcal{Z}^s \setminus \mathcal{Z}$ and $\hat{\mathcal{Z}}^t = \mathcal{Z}^t \setminus \mathcal{Z}$ represent domain-private labels. UniDA assumes no prior knowledge of $\mathcal{Z}^t$, $\mathcal{Z}$, or $\hat{\mathcal{Z}}^t$, aiming to identify shared-label samples while rejecting private-label instances. In source-free setting, $\mathcal{S}$ is only available during pre-training. Given a source model $p_\phi^s = q_\phi^s \circ r_\phi^s$, where $r_\phi^s : \mathbb{R}^d \to \mathbb{R}^m$ extracts features and $q_\phi^s : \mathbb{R}^m \to \mathbb{R}^K$ classifies them, adaptation involves learning a target encoder r_ϕ^t while maintaining $q_\phi^t = q_\phi^s$. Figure 1 shows ESBN.

Batch normalization. For convolutional neural networks (CNNs), the batch normalization (BN) layer processes four-dimensional tensors. Let $X_{n,k,h,w}$ and $Y_{n,k,h,w}$ denote the input and output tensors respectively, where n indexes samples within a batch, k represents feature channels, and (h, w) indicates spatial coordinates. In the case of input images, k typically corresponds to the RGB color channels. The normalization is performed independently within each channel:

$$Y_{n,k,h,w} = \text{BN}(\mathbf{X}_{\mathbf{n},\mathbf{k},\mathbf{h},\mathbf{w}}) = \alpha_k \cdot \frac{X_{n,k,h,w} - m_k}{\sqrt{v_k + \delta}} + \theta_k, \quad (1)$$

where $m_k = \frac{1}{|\mathcal{M}|} \sum_{n,h,w} X_{n,k,h,w}$ represents the mean activation in channel k, computed across all samples n in the mini-batch and all spatial locations (h, w), where $\mathcal{M}$ denotes the set of all activations used in the mean computation. The variance v_k is calculated analogously. A small constant δ is added for numerical stability. During inference, running averages of means and variances accumulated during training are utilized. Following normalization, a channel-wise affine transformation is applied using learnable parameters α_k and θ_k, which are optimized during the training process. The transformation effectively standardizes the distribution of activations within each channel while maintaining the network's representational capacity through the learnable parameters. This standardization helps mitigate internal covariate shift and enables more stable training of deep neural networks.

Batch-free Normalization. Batch-free normalization (BFN) techniques are designed to eliminate the dependency on batch statistics, ensuring consistent behavior during both training and inference phases. Layer normalization (LN) [Lei Ba *et al.*, 2016] operates by computing statistics across all feature channels and spatial dimensions for each individual sample independently, which can be formulated as:

$$Y_{n,k,h,w} = \text{LN}(\mathbf{X}_{\mathbf{n},\mathbf{k},\mathbf{h},\mathbf{w}}) = \alpha \cdot \frac{X_{n,k,h,w} - \mu_n}{\sqrt{\sigma_n^2 + \delta}} + \theta, \quad (2)$$

where $\mu_n = \frac{1}{CHW} \sum_{k,h,w} X_{n,k,h,w}$ computes the mean across channels (C) and spatial dimensions (H, W) for each sample n. The variance σ_n^2 is computed similarly. Unlike BN, the learnable parameters α and θ are shared across all channels. Group normalization (GN) [Wu and He, 2018] extends this concept by partitioning channels into G groups and performing normalization within each group separately:

$$Y_{n,k,h,w} = \text{GN}(\mathbf{X}_{\mathbf{n},\mathbf{k},\mathbf{h},\mathbf{w}}) = \alpha_g \cdot \frac{X_{n,k,h,w} - \mu_{n,g}}{\sqrt{\sigma_{n,g}^2 + \delta}} + \theta_g, \quad (3)$$

$$\begin{aligned} Y_{n,k,h,w} &= \text{GN}(\mathbf{X}_{\mathbf{n},\mathbf{k},\mathbf{h},\mathbf{w}}) \\ &= \alpha_g \cdot \frac{X_{n,k,h,w} - \mu_{n,g}}{\sqrt{\sigma_{n,g}^2 + \delta}} + \theta_g, \end{aligned} \quad (4)$$

where $\mathcal{G}g$ represents the set of channels in group g, and $\mu n, g$ and $\sigma_{n,g}^2$ are computed within each group independently. This design offers a flexible trade-off between LN ($G = 1$) and instance normalization ($G = C$), making it particularly effective for visual tasks where small batch training is prevalent.

3.2 Estimation Shift of Batch Normalization

Consider a training dataset $\mathcal{D}$ and its mini-batches $\{D^i\}_{i=1}^N$. For a neural network with BN layer $G_{\phi,\omega}(D) = G_\phi^{post}(BN(G_\omega^{pre}(D)))$, let $Z = G_\omega^{pre}(D)$ and $\hat{Z} = BN(Z)$. While the population statistics of $\mathcal{D}$ are well-defined through mini-batch statistics, those of activations $\mathbf{Z} = G_\omega^{pre}(\mathcal{D})$ are challenging to define due to parameter updates during training. Specifically, mini-batch activations $Z^i = G_{\omega^i}^{pre}(D^i)$ depend on both input D^i and evolving model parameters $\{\omega^i\}_{i=1}^N$. Despite this complexity, we observe that $\hat{Z}^i$ maintains standardized distribution across iterations. We define the expected population statistics: for a trained model $G_{\bar{\phi},\bar{\omega}}(\cdot)$ on training data $\mathcal{D}$, the expected population statistics $\bar{\mu}, \bar{\sigma}^2$ of BN are defined as the mean and variance of activations $\mathbf{Z} = G_{\bar{\omega}}^{pre}(\mathcal{D}')$ on test data $\mathcal{D}'$ [Huang *et al.*, 2022].

According to the expected population statistics of BN, we define the Estimation Shift of BN as the discrepancy between the estimated population statistics and the expected ones. Understanding the impact of the Estimation Shift on the performance of the batch-normalized network is crucial. Therefore, we aim to reduce the difference between the estimated and expected statistics, thus improving the generalization performance of the network model.

3.3 BFN in Domain Adaptation

To estimate the distribution of feature magnitudes $\|\mathbf{v}_{i,n}\|_2$, ESBN uses a two-component Gaussian Mixture Model (GMM) [Qu *et al.*, 2024a]. The components with lower and higher means, denoted as θ_s and θ_n respectively where $\theta_s < \theta_n$, correspond to shared and novel classes. To achieve accurate class separation, we construct class-specific reference vectors through top-M sampling. For each shared class, we maintain two types of references: target references $\{\mathbf{r}_k^t \in \mathbb{R}^D|_{k=1,\ldots,K}\}$ and source references $\{\mathbf{r}_k^s \in \mathbb{R}^D|_{k=1,\ldots,K}\}$. The sampling size M is determined by $M = N_t/\hat{K}_t$, where N_t represents target samples and $\hat{K}_t$ denotes estimated target classes. The source references are derived from classifier weights to handle potential source-novel scenarios.

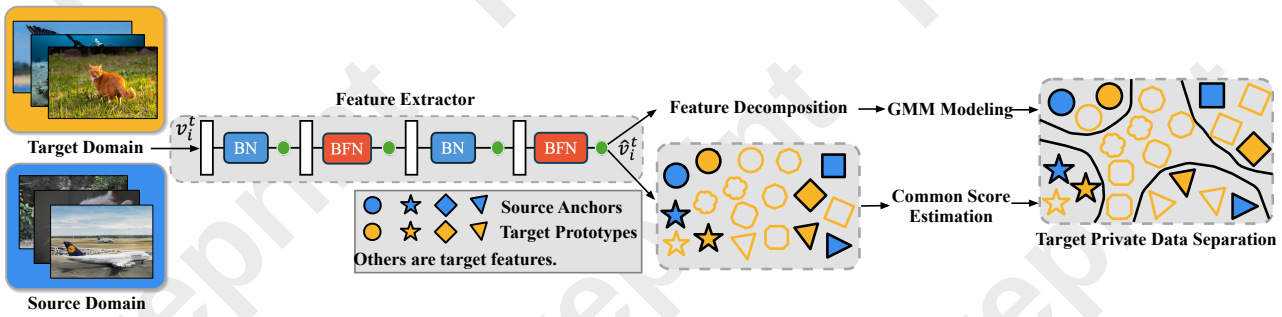


Figure 1: Overall framework of ESBN. BN and BFN are mixed in different layers.

Before feeding the features into the GMM for class separation, we apply BFN to the feature vectors $\mathbf{v}_i$ to ensure their mean and variance are standardized within each instance. Unlike BN, BFN does not require batch-wise statistics, and instead, it normalizes across the channels of each individual sample. This method makes it particularly useful when handling domain adaptation tasks, as it alleviates issues related to batch size and domain-specific statistics. The BFN for a feature vector $\mathbf{v}i \in \mathbb{R}^C$ (with C channels) can be expressed as:

$$\hat{\mathbf{v}}i = \gamma \cdot \frac{\mathbf{v}i - \mu_i}{\sigma_i + \epsilon} + \beta, \tag{5}$$

where $\mu_i = \frac{1}{C}\sum c = 1^C vi, c$ is the mean of the feature vector $\mathbf{v}i$, $\sigma_i = \sqrt{\frac{1}{C}\sum c = 1^C (vi, c - \mu_i)^2}$ is the standard deviation, and γ and β are learnable scaling and shifting parameters. This normalization ensures that the feature vector for each instance has zero mean and unit variance, making it suitable for domain adaptation tasks where features from both source and target domains are expected to have varying distributions. Based on these references, we define a confidence metric $\alpha_{i,k}$ that evaluates instance-to-class relationships:

$$\begin{aligned} s(\mathbf{p}, \mathbf{q}) &= 1.0 - \cos(\mathbf{p}, \mathbf{q}), \\ \alpha^t_{i,k} &= 1.0 - \exp(s(\mathbf{v}^t_i, \mathbf{r}^t_k) - 1.0), \\ \alpha^s_{i,k} &= \exp(-s(\mathbf{v}^t_i, \mathbf{r}^s_k)), \\ \alpha_{i,k} &= \sqrt{\alpha^t_{i,k} \cdot \alpha^s_{i,k}}, \end{aligned} \tag{6}$$

in addition to the normalization of features, BFN ensures that the feature distributions are stable within each instance, which is particularly helpful in handling small batch sizes or domain shift. The confidence scores $\alpha^t_{i,k}$ and $\alpha^s_{i,k}$ are clipped to [0, 1] before computing their geometric mean. The asymmetric formulation ensures that proximity to source references carries more weight in shared class determination than proximity to target references. The class assignment combines distribution statistics with instance-specific confidence:

$$\begin{aligned} \beta_{i,k} &= \theta_k + \alpha_{i,k}(\theta_n - \theta_k), \\ \hat{\mathbf{y}}^t_i &= \begin{cases} \text{novel}, & \|\mathbf{v}^t_{i,n}\|_2 \geq \beta_{i,k}, \\ \mathbf{1}(\operatorname{argmax} \alpha_i), & \|\mathbf{v}^t_{i,n}\|_2 < \beta_{i,k}. \end{cases} \end{aligned} \tag{7}$$

Here, $\mathbf{v}^t_{i,n}$ for novel data, θ_k represents the expected magnitude for class k obtained through the same top-M sampling procedure, and $\beta_{i,k}$ defines an instance-specific decision boundary that accounts for both class-level distribution and instance-level confidence. $\mathbf{1}$ denotes the one-hot encoding operator.

3.4 Optimization Objectives

The training process incorporates three complementary objectives to achieve effective domain adaptation: pseudo-label learning, feature magnitude regularization, and distribution alignment. Firstly, based on the predicted labels from the adaptive thresholding mechanism, we employ a weighted cross-entropy loss for model optimization. To account for prediction reliability, we introduce a confidence weight using a Student's t-distribution that considers the distance between feature magnitude $|\mathbf{v}^t_{i,n}|2$ and decision boundary $\beta i, k$:

$$w^t_i \propto 1 - \left(1 + \frac{(\beta_{i,k} - |\mathbf{v}^t_{i,n}|2)^2}{\eta}\right)^{-\frac{\eta+1}{,}2}, \tag{8}$$

$$\begin{aligned} \mathcal{L}cls = \\ -\frac{1}{N}\sum_{i=1}^{N} w^t_i \cdot \sum_{k=1}^{K} \hat{\mathbf{y}}^t_{i,k} \log p_k(\mathbf{x}^t_i), \end{aligned} \tag{9}$$

where w^t_i represents the sample confidence, $p_k(\mathbf{x}^t_i)$ denotes the predicted probability for class k, and η is set to 1e-4. For samples classified as novel, we use a uniform distribution over shared classes instead of introducing an additional category. Secondly, to enhance the discriminative power of feature magnitudes for novel class detection, we introduce a regularization term:

$$q_i = \frac{\exp(|\mathbf{v}^t_{i,n}|2)}{\exp(|\mathbf{v}^t i, n|2) + \exp(|\mathbf{v}^t i, s|2)}, \tag{10}$$

$$\begin{aligned} \mathcal{L}mag = \\ -\frac{1}{N}\sum_{i=1}^{N} w^t_i \cdot \left(z^t_i \log(q_i) + (1 - z^t_i)\log(1 - q_i)\right), \end{aligned} \tag{11}$$

where z^t_i is a binary indicator (1 for novel, 0 for shared classes), $\mathbf{v}^t_{i,s}$ for shared data. Finally, to ensure stable feature

Methods	U	SF	CF	Ar2Cl	Ar2Pr	Ar2Re	Cl2Ar	Cl2Pr	Cl2Re	Pr2Ar	Pr2Cl	Pr2Re	Re2Ar	Re2Cl	Re2Pr	Avg
CMU [Fu *et al.*, 2020]	×	×	✓	56.0	56.9	59.2	67.0	64.3	67.8	54.7	51.1	66.4	68.2	57.9	69.7	61.6
DANCE [Saito *et al.*, 2020]	✓	×	✓	61.0	60.4	64.9	65.7	58.8	61.8	73.1	61.2	66.6	67.7	62.4	63.7	63.9
DCC [Li *et al.*, 2021]	✓	×	×	58.0	54.1	58.0	**74.6**	70.6	77.5	64.3	**73.6**	74.9	**81.0**	**75.1**	80.4	70.2
OVANet [Saito and Saenko, 2021]	×	×	✓	**62.8**	75.6	78.6	70.7	68.8	75.0	71.3	58.6	80.5	76.1	64.1	78.9	71.8
Source-only	-	-	-	47.3	71.6	81.9	51.5	57.2	69.4	56.0	40.3	76.6	61.4	44.2	73.5	60.9
SHOT [Liang *et al.*, 2020]	×	✓	✓	32.9	29.5	39.6	56.8	30.1	41.1	54.9	35.4	42.3	58.5	33.5	33.3	40.7
GLC [Qu *et al.*, 2023]	✓	✓	✓	55.1	**78.1**	**89.2**	63.1	**80.5**	**89.1**	**77.6**	53.8	**88.9**	**80.6**	54.4	**85.9**	**74.7**
LEAD [Qu *et al.*, 2024a]	✓	✓	✓	61.1	78.0	86.4	70.6	73.8	**83.4**	75.3	59.4	86.0	74.3	60.7	**83.6**	74.4
ESBN	✓	✓	✓	**64.7**	**78.4**	**89.0**	**71.7**	**74.0**	80.0	**76.5**	**62.0**	**87.5**	76.5	**65.1**	82.4	**75.7**

Table 1: H-score (%) averaged on **Office-Home** for OPDA scenario. U denotes methods applicable for all potential label-shift scenarios. SF represents source data-free. CF indicates model adaptation without time-consuming K-means clustering.(Best in **red** and second best in **blue**.)

learning, we incorporate a distribution alignment loss $\mathcal{L}_{align}$ that encourages consistency between neighboring samples in feature space. This helps maintain consistent predictions for semantically similar instances across domains. The final optimization objective combines these components:

$$\mathcal{L}total = \gamma \cdot \mathcal{L}cls + \mathcal{L}mag + \mathcal{L}align, \tag{12}$$

where γ is a positive balancing coefficient. This combined objective enables effective separation of shared and novel classes while maintaining discriminative feature learning.

4 Experiments

4.1 Experimental Setup

Datasets. ***Office-31*** [Saenko *et al.*, 2010] is a widely used DA benchmark comprising three domains: Amazon (A), Dslr (D), and Webcam (W), with 31 office object classes. ***Office-Home*** [Venkateswara *et al.*, 2017] includes four domains: Artistic (Ar), Clip Art (Cl), Product (Pr), and Real-world (Re), covering 65 object classes. ***VisDA-C*** [Peng *et al.*, 2017] is a large-scale DA dataset with 12 classes across synthetic and real domains, focusing on synthetic-to-real transfer. Class splits for each setting are summarized in Table 6.

Implementation Details. All experiments are implemented on a single RTX-3090 GPU. Existing baselines maintain their original settings, with the feature encoder fine-tuned from the ImageNet pre-trained ResNet-50 backbone. The model is trained using SGD with a momentum of 0.9 and weight decay of 1×10^{-3}. The learning rate is set to 1×10^{-3} for the Office-31 and Office-Home datasets, and 1×10^{-4} for the VisDA-C datasets. The batch size is 64. Apart from the parameters introduced for our improvements, all other settings remain consistent with those in [Qu *et al.*, 2024a]. The source codes are available at https://github.com/leuluelueluelue/ESBN.

Evaluation. We adopt the same evaluation metrics as prior works for a fair comparison. In PDA scenarios, we report the classification accuracy as the primary evaluation metric. For OSDA and OPDA scenarios, we utilize the H-score [Liang *et al.*, 2021] to evaluate performance. The H-score is defined as the harmonic mean of accuracy over common data and accuracy over private data.

4.2 Comparative Results

Results for OPDA Scenario. Experiments were initially conducted under the most challenging OPDA scenario to evaluate the effectiveness of the proposed method. Table 1 shows the results on the Office-Home dataset, while Table 2 summarizes the outcomes on the Office-31 and VisDA datasets. These results evidence that ESBN outperforms these existing methods in effectively segregating common and private data, highlighting its superior capability in handling DA tasks. Specifically, on the Office-Home, Office-31, and VisDA datasets, ESBN surpasses the state-of-the-art baseline by 1%, 0.9%, and 1.6%, respectively. We attribute this improvement to the effective implementation of the proposed method, which effectively mitigates the limitations associated with traditional BN. In particular, we replace the BN layers in even/odd layers within the residual blocks of ResNet with BFN.

Results for OSDA Scenario. In the OSDA scenario only the target domain involves private data, necessitating a method that can effectively differentiate between common and private data. The H-score results presented in Table 3 and Table 5 illustrate the performance of various methods under this scenario on the Office-Home, Office-31, VisDA datasets. Notably, ESBN demonstrates superior performance on the Office-Home, Office-31, and VisDA datasets, achieving improvements of 1.2%, 1.1%, and 1.4% over the LEAD method, respectively. It also maintains a leading position among all baselines, similar to the observations in the OPDA scenarios.

Results for PDA Scenario. Finally, we validate the effectiveness of ESBN in the PDA scenario. The results shown in Table 4 and Table 5 indicate that ESBN achieves excellent performance even when compared to methods specifically tailored for PDA. In particular, ESBN attains overall accuracies of 72.7%, 95.5%, and 76.2% on the Office-Home, Office-31 and VisDA datasets, respectively. It is noteworthy that the accuracy on the Office-Home dataset is slightly lacking, which may be attributed to gradient propagation issues caused by modifications to the BN layer, leading to compromises during optimization. We will seek better modification and integration strategies in our future work.

4.3 Experimental Analysis

Ablation Study. To study the impact of normalization layer modifications in ESBN , we perform ablations on three datasets in the OSDA scenario. Results are shown in Tables 7 and 8, with Table 7 highlighting the effect of different normalization strategies. The results show that substituting BN layers with BFN in the even layers within even residual locks (EE) yields the best performance. Specifically, ESBN achieves 66.6%, 91.3%, and 73.6% on the Office-Home, Office-31, and VisDA datasets, respectively. We at-

Methods	U	SF	CF	Office-31 A2D	A2W	D2A	D2W	W2A	W2D	Avg	VisDA S2R
CMU [Fu *et al.*, 2020]	×	×	✓	68.1	67.3	71.4	79.3	72.2	80.4	73.1	32.9
DANCE [Saito *et al.*, 2020]	✓	×	✓	78.6	71.5	79.9	91.4	72.2	87.9	80.3	42.8
DCC [Li *et al.*, 2021]	✓	×	×	**88.5**	78.5	70.2	79.3	75.9	88.6	80.2	43.0
OVANet [Saito and Saenko, 2021]	×	×	✓	**85.8**	79.4	80.1	**95.4**	84.0	**94.3**	86.5	53.1
Source-only	-	-	-	70.9	63.2	39.6	77.3	52.2	86.4	64.9	25.7
SHOT [Liang *et al.*, 2020]	×	✓	✓	73.5	67.2	59.3	88.3	77.1	84.4	75.0	44.0
GLC [Qu *et al.*, 2023]	✓	✓	✓	81.5	**84.5**	**89.8**	90.4	**88.4**	92.3	**87.8**	73.1
LEAD [Qu *et al.*, 2024a]	✓	✓	✓	83.7	**85.0**	86.3	90.9	86.2	93.1	87.5	**76.5**
ESBN	✓	✓	✓	85.3	83.9	**86.9**	**94.0**	**87.3**	**94.6**	**88.7**	**78.1**

Table 2: H-score (%) comparison in OPDA scenario on Office-31 and VisDA datasets.

Methods	U	SF	CF	Ar2Cl	Ar2Pr	Ar2Re	Cl2Ar	Cl2Pr	Cl2Re	Pr2Ar	Pr2Cl	Pr2Re	Re2Ar	Re2Cl	Re2Pr	Avg
CMU [Fu *et al.*, 2020]	×	×	✓	55.0	57.0	59.0	59.3	58.2	60.6	59.2	51.3	61.2	61.9	53.5	55.3	57.6
DANCE [Saito *et al.*, 2020]	✓	×	✓	6.5	9.0	9.9	20.4	10.4	9.2	28.4	12.8	12.6	14.2	7.9	13.2	12.9
DCC [Li *et al.*, 2021]	✓	×	×	56.1	67.5	66.7	49.6	66.5	64.0	55.8	53.0	70.5	61.6	57.2	71.9	61.7
OVANet [Saito and Saenko, 2021]	×	×	✓	58.6	66.3	69.9	**62.0**	65.2	68.6	59.8	53.4	69.3	**68.7**	**59.6**	66.7	64.0
Source-only	-	-	-	46.1	63.3	72.9	42.8	54.0	58.7	47.8	36.1	66.2	60.8	45.3	68.2	55.2
SHOT [Liang *et al.*, 2020] [27]	×	✓	✓	37.7	41.8	48.4	56.4	39.8	40.9	60.0	41.5	49.7	61.8	41.4	43.6	46.9
GLC [Qu *et al.*, 2023]	✓	✓	✓	**63.9**	**74.2**	**77.9**	59.8	**70.2**	**73.9**	61.1	**63.0**	**75.8**	**65.7**	**63.6**	**77.6**	**68.9**
LEAD [Qu *et al.*, 2024a]	✓	✓	✓	**60.0**	70.4	**76.5**	**61.0**	66.9	**70.8**	**63.5**	**55.2**	74.0	64.2	51.7	75.5	65.8
ESBN	✓	✓	✓	59.7	**79.0**	72.6	57.6	**75.0**	69.6	**65.6**	50.2	**77.8**	62.8	57.4	**77.1**	**67.0**

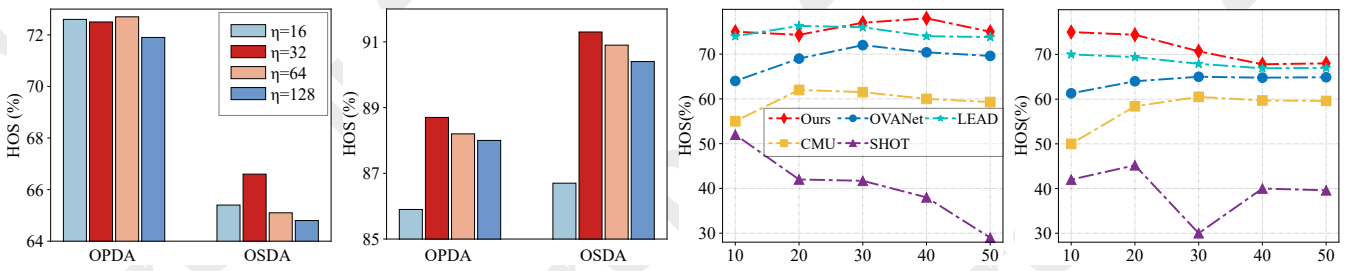
Table 3: H-score (%) comparison in OSDA scenario on Office-Home.

Methods	U	SF	CF	Ar2Cl	Ar2Pr	Ar2Re	Cl2Ar	Cl2Pr	Cl2Re	Pr2Ar	Pr2Cl	Pr2Re	Re2Ar	Re2Cl	Re2Pr	Avg
CMU [Fu *et al.*, 2020]	×	×	✓	50.9	74.2	78.4	62.2	64.1	72.5	63.5	47.9	78.3	72.4	54.7	78.9	66.5
DANCE [Saito *et al.*, 2020]	✓	×	✓	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
DCC [Li *et al.*, 2021]	✓	×	×	54.2	47.5	57.5	**83.8**	71.6	**86.2**	63.7	**65.0**	75.2	**85.5**	**78.2**	82.6	70.9
OVANet [Saito and Saenko, 2021]	×	×	✓	34.1	54.6	72.1	42.4	47.3	55.9	38.2	26.2	61.7	56.7	35.8	68.9	49.5
Source-only	-	-	-	45.9	69.2	81.1	55.7	61.2	64.8	60.7	41.1	75.8	70.5	49.9	78.4	62.9
SHOT [Liang *et al.*, 2020]	×	✓	✓	**64.7**	**85.1**	**90.1**	**75.1**	**73.9**	84.2	**76.4**	**64.1**	**90.3**	**80.7**	**63.3**	**85.5**	**77.8**
GLC [Qu *et al.*, 2023]	✓	✓	✓	55.9	79.0	**87.5**	72.5	71.8	82.7	**74.9**	41.7	82.4	77.3	60.4	84.3	72.5
LEAD [Qu *et al.*, 2024a]	✓	✓	✓	**56.7**	**82.0**	86.5	70.0	**75.2**	80.7	73.5	47.2	82.2	78.0	57.5	80.0	**72.4**
ESBN	✓	✓	✓	56.0	76.6	86.5	71.8	71.4	**84.3**	73.2	45.6	83.6	80.6	52.0	**84.6**	72.2

Table 4: Accuracy comparison (%) in PDA scenario on Office-Home.

Methods	U	SF	CF	Office-31 A2D	A2W	D2A	D2W	W2A	W2D	Avg	VisDA S2R	Office-31 A2D	A2W	D2A	D2W	W2A	W2D	Avg	VisDA S2R
CMU [Fu *et al.*, 2020]	×	×	✓	52.6	55.7	76.5	75.9	65.8	64.7	65.2	54.2	84.1	84.2	69.2	97.2	66.8	98.8	83.4	65.5
DANCE [Saito *et al.*, 2020]	✓	×	✓	84.9	78.8	79.1	78.8	68.3	78.8	78.1	67.5	77.1	71.2	83.7	94.6	92.6	96.8	86.0	73.7
DCC [Li *et al.*, 2021]	✓	×	×	58.3	54.8	67.2	89.4	85.3	80.9	72.7	59.6	87.3	81.3	**95.4**	**100.0**	95.5	100.0	93.3	72.4
OVANet [Saito and Saenko, 2021]	×	×	✓	**90.5**	**88.3**	86.7	**98.2**	88.3	**98.4**	**91.7**	66.1	69.4	61.7	61.4	90.2	66.4	98.7	74.6	34.3
Source-only	-	-	-	78.2	72.1	44.2	82.2	52.1	88.8	69.6	29.1	**90.4**	79.3	79.3	95.9	84.3	98.1	87.8	42.8
SHOT [Liang *et al.*, 2020]	×	✓	✓	80.2	71.6	64.3	93.1	64.0	91.8	77.5	28.1	89.8	84.4	92.2	96.6	92.2	99.4	92.4	74.2
GLC [Qu *et al.*, 2023]	✓	✓	✓	82.6	74.6	**92.6**	**96.0**	**91.8**	96.1	89.0	**72.5**	**89.8**	89.8	92.8	96.6	**96.1**	99.4	94.1	**76.0**
LEAD [Qu *et al.*, 2024a]	✓	✓	✓	84.9	85.1	**90.1**	94.5	**90.3**	96.1	90.2	72.2	89.8	**93.9**	93.8	98.3	95.9	**99.4**	**95.2**	75.2
ESBN	✓	✓	✓	**89.5**	**91.3**	89.0	95.6	88.7	**96.6**	**91.8**	**73.6**	88.5	**94.6**	**95.5**	**98.6**	**96.0**	**100.0**	**95.5**	**76.2**

Table 5: H-score (%) in OSDA scenario and Accuracy comparison (%) in PDA scenario on Office-31 and VisDA. On the left are the results of the experiments in the OSDA scenario and on the right are the results of the experiments in the PDA scenario.



(a) Office-Home (b) Office-31 (c) Cl→Pr on OPDA (d) Cl→Pr on OSDA

Figure 2: (a-b) shows the sensitivity to η on Office-home and Office-31. (c-d) present robustness analysis when varying the unknown private categories. HOS stands for H-score.

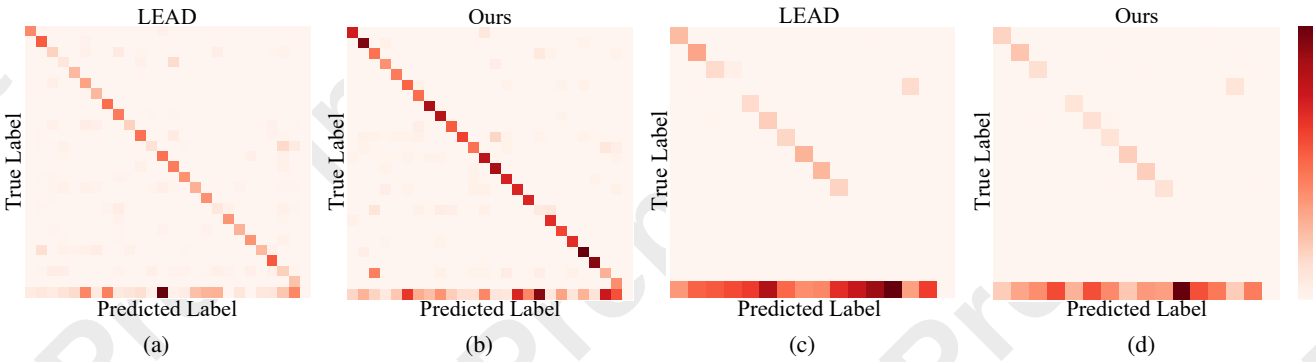


Figure 3: The confusion matrices for LEAD and ESBN in the OSDA and OPDA scenarios Cl→Ar on the Office-Home dataset. (a-b) shows on the OSDA scenario. (c-d) shows on the OPDA scenario.

Dataset	Class Splits($\mathcal{C}/\bar{\mathcal{C}}_s/\bar{\mathcal{C}}_t$)		
	OPDA	OSDA	PDA
Office-Home	10/5/50	25/0/40	25/40/0
Office-31	10/10/11	10/0/11	10/21/0
VisDA	6/3/3	6/0/6	6/6/0

Table 6: Details of class splits for each setting. $\mathcal{C}$: the common (shared) classes across domains, $\bar{\mathcal{C}}_s$: the source-private (novel) classes, and $\bar{\mathcal{C}}_t$: the target-private (novel) classes.

Layer Strategy	Office-Home	Office-31	VisDA
w/o GN	65.8	90.2	72.2
AO	59.3	85.2	64.2
AE	63.4	78.5	59.5
OO	65.0	89.2	62.9
EE	**66.6**	**91.3**	**73.6**

Table 7: Ablation study on the impact of BFN layers on the model in OSDA scenarios. **EE****OO** refers to the modification of BN layers in even\odd layers within even\odd ResidualBlocks of ResNet, respectively. **AE****AO** refers to the modification of BN layers in even\odd layers within all ResidualBlocks of ResNet, respectively. By default, these BN layers are substituted with GN.

tribute this improvement to BFN, which enhances the stability of training and enables the model to better adapt to the data distributions across different domains. In contrast, the other strategies do not lead to significant performance improvements, particularly in the context of more complex DA tasks. Table 8 further corroborates that GN is the most effective BFN strategy for improving DA performance.

Parameter sensitivity. Figure 2 (a-b) shows how varying η affects model performance. On both Office-Home and Office-31 datasets, $\eta = 32$ consistently achieves the highest H-score, indicating improved model adaptability.

Robustness. We primarily conduct the robustness analysis of ESBN from the perspective of varying unknown classes. As the number of "unknown" classes increases, it becomes increasingly challenging to accurately distinguish between "unknown" and "known" objects. However, as shown in Figure 2 (c-d), our model exhibits excellent stability.

Visualization analysis. Based on the analysis of the confusion matrices in Figure 3, it is evident that ESBN significantly outperforms the LEAD in both OSDA and OPDA scenarios. ESBN achieves a higher proportion of correct classifications, reflecting its high accuracy in distinguishing between known and unknown categories. Moreover, the relatively low occurrence of misclassifications demonstrates the model's relatively strong category discrimination capability. Additionally, ESBN exhibits excellent stability, maintaining consistent performance across both OSDA and OPDA scenarios.

BFN Strategy	Office-Home	Office-31	VisDA
BN	65.8	90.2	72.2
IN	63.4	90.9	71.4
LN	65.1	89.4	67.6
GN	**66.6**	**91.3**	**73.6**

Table 8: Ablation study on the impact of BFN strategy on the model in OSDA scenarios.

5 Conclusion

This paper presents an investigation into the strategic use of BFN for domain adaptation. We address a key gap in existing research by exploring how the placement and depth of normalization layers affect cross-domain generalization. Through extensive experiments, we show that selectively replacing BN with BFN at specific network positions significantly enhances generalization performance while preserving feature discriminability. Our findings highlight the critical role of normalization layer placement, with BFN providing more robust feature representations by computing instance-specific statistics, particularly in the presence of domain shifts. ESBN outperforms traditional domain adaptation techniques across multiple benchmarks, demonstrating its practical applicability in real-world settings. This work offers promising directions for improving domain adaptation in complex and real-world environments.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grants No. 62406100, Tianjin Natural Science Foundation under Grants No. 24JCQNJC00320, Beijing Postdoctoral Research Foundation. This research is

also supported by funding from the Hong Kong RGC General Research Fund (No. 152211/23E, 15216424/24E, and 152115/25E), the PolyU Internal Fund (No. P0056171), and the Huawei Gifted Fund.

Contribution Statement

Jiao Li and Houcheng Su contributed equally to this work.

References

[Cao *et al.*, 2019] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Du *et al.*, 2024] Yuntao Du, Haiyang Yang, Mingcai Chen, Hongtao Luo, Juan Jiang, Yi Xin, and Chongjun Wang. Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. *Machine Learning*, 113(6):3611–3631, 2024.

[Fang *et al.*, 2024] Yuqi Fang, Pew-Thian Yap, Weili Lin, Hongtu Zhu, and Mingxia Liu. Source-free unsupervised domain adaptation: A survey. *Neural Networks*, page 106230, 2024.

[Fu *et al.*, 2020] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 567–583, 2020.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, pages 2096–2030, 2016.

[Hoffman *et al.*, 2018] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 1989–1998, 2018.

[Hou and Zheng, 2020] Yunzhong Hou and Liang Zheng. Source free domain adaptation with image translation. *arXiv preprint arXiv:2008.07514*, 2020.

[Huang *et al.*, 2022] Lei Huang, Yi Zhou, Tian Wang, Jie Luo, and Xianglong Liu. Delving into the estimation shift of batch normalization in a network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 763–772, 2022.

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

[Li *et al.*, 2021] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[Li *et al.*, 2024] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. A comprehensive survey on source-free domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[Liang *et al.*, 2020] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

[Liang *et al.*, 2021] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Umad: Universal model adaptation under domain and category shift. *arXiv preprint arXiv:2112.08553*, 2021.

[Liu and Zhou, 2024] Xinghong Liu and Yi Zhou. Coca: Classifier-oriented calibration via textual prototype for source-free universal domain adaptation. In *Proceedings of the Asian Conference on Computer Vision*, pages 1671–1687, 2024.

[Liu *et al.*, 2019] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2927–2936, 2019.

[Liu *et al.*, 2021] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1215–1224, 2021.

[Peng *et al.*, 2017] 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.

[Peng *et al.*, 2019] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[Qiu *et al.*, 2021] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *arXiv preprint arXiv:2106.15326*, 2021.

[Qu *et al.*, 2022] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[Qu *et al.*, 2023] Sanqing Qu, Tianpei Zou, Florian Röhrbein, Cewu Lu, Guang Chen, Dacheng Tao, and Changjun Jiang. Upcycling models under domain and category shift. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[Qu *et al.*, 2024a] Sanqing Qu, Tianpei Zou, Lianghua He, Florian Röhrbein, Alois Knoll, Guang Chen, and Changjun Jiang. Lead: Learning decomposition for source-free universal domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[Qu *et al.*, 2024b] Sanqing Qu, Tianpei Zou, Florian Röhrbein, Cewu Lu, Guang Chen, Dacheng Tao, and Changjun Jiang. Glc++: Source-free universal domain adaptation through global-local clustering and contrastive affinity learning. *arXiv preprint arXiv:2403.14410*, 2024.

[Rangwani *et al.*, 2022] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *International Conference on Machine Learning (ICML)*, pages 18378–18399, 2022.

[Saenko *et al.*, 2010] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[Saito and Saenko, 2021] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[Saito *et al.*, 2018] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3723–3732, 2018.

[Saito *et al.*, 2020] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. In *Advances in Neural Information Processing systems (Neurips)*, pages 16282–16292, 2020.

[Sankaranarayanan *et al.*, 2018] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8503–8512, 2018.

[Schlachter and Yang, 2024] Pascal Schlachter and Bin Yang. Comet: Contrastive mean teacher for online source-free universal domain adaptation. *arXiv preprint arXiv:2401.17728*, 2024.

[Tranheden *et al.*, 2021] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)*, pages 1379–1389, 2021.

[Tzeng *et al.*, 2017] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017.

[Venkateswara *et al.*, 2017] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Wang *et al.*, 2024a] Mengzhu Wang, Junze Liu, Ge Luo, Shanshan Wang, Wei Wang, Long Lan, Ye Wang, and Feiping Nie. Smooth-guided implicit data augmentation for domain generalization. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Wang *et al.*, 2024b] Mengzhu Wang, Yuehua Liu, Jianlong Yuan, Shanshan Wang, Zhibin Wang, and Wei Wang. Inter-class and inter-domain semantic augmentation for domain generalization. *IEEE Transactions on Image Processing*, 33:1338–1347, 2024.

[Wang *et al.*, 2024c] Mengzhu Wang, Shanshan Wang, Xun Yang, Jianlong Yuan, and Wenju Zhang. Equity in unsupervised domain adaptation by nuclear norm maximization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7):5533–5545, 2024.

[Wang *et al.*, 2024d] Yunyun Wang, Jian Mao, Cong Zou, and Xinyang Kong. Universal domain adaptation from multiple black-box sources. *Image and Vision Computing*, 142:104896, 2024.

[Wu and He, 2018] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[Wu *et al.*, 2024] Yanan Wu, Zhixiang Chi, Yang Wang, Konstantinos N Plataniotis, and Songhe Feng. Test-time domain adaptation by learning domain-aware batch normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15961–15969, 2024.

[Xiao *et al.*, 2021] Teng Xiao, Zhengyu Chen, Donglin Wang, and Suhang Wang. Learning how to propagate messages in graph neural networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1894–1903, 2021.

[Yang *et al.*, 2021a] Shiqi Yang, Joost Van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in Neural Information Processing systems (Neurips)*, 34:29393–29405, 2021.

[Yang *et al.*, 2021b] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8978–8987, 2021.

[You *et al.*, 2019] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2729, 2019.

[Zellinger *et al.*, 2017] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[Zhang *et al.*, 2019] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5031–5040, 2019.