On Multi-Domain Long-Tailed Recognition, Imbalanced Domain Generalization and Beyond

Yuzhe Yang[1], Hao Wang[2], and Dina Katabi[1]

[1] MIT CSAIL [2] Rutgers University

Abstract. Real-world data often exhibit imbalanced label distributions. Existing studies on data imbalance focus on single-domain settings, i.e., samples are from the same data distribution. However, natural data can originate from distinct domains, where a minority class in one domain could have abundant instances from other domains. We formalize the task of Multi-Domain Long-Tailed Recognition (MDLT), which learns from multi-domain imbalanced data, addresses *label imbalance*, *domain shift*, and *divergent label distributions across domains*, and generalizes to all domain-class pairs. We first develop the *domain-class transferability graph*, and show that such transferability governs the success of learning in MDLT. We then propose `BoDA`, a theoretically grounded learning strategy that tracks the upper bound of transferability statistics, and ensures *balanced* alignment and calibration across imbalanced domain-class distributions. We curate five MDLT benchmarks based on widely-used multi-domain datasets, and compare `BoDA` to twenty algorithms that span different learning strategies. Extensive and rigorous experiments verify the superior performance of `BoDA`. Further, as a byproduct, `BoDA` establishes new state-of-the-art on Domain Generalization benchmarks, highlighting the importance of addressing data imbalance across domains, which can be crucial for improving generalization to unseen domains. Code and data are available at: `https://github.com/YyzHarry/multi-domain-imbalance`.

1 Introduction

Real-world data often exhibit label imbalance – i.e., instead of a uniform label distribution over classes, in reality, data are by their nature imbalanced: a few classes contain a large number of instances, whereas many others have only a few instances [5,6,52]. This phenomenon poses a challenge for deep recognition models, and has motivated several prior solutions [6,10,33,39,52,53]. Such prior solutions focus on *single domain* scenarios, i.e., samples are from the same data distribution; they propose techniques for learning from imbalanced training data and generalizing to a balanced test set.

In contrast, this paper formulates the problem of *Multi-Domain Long-Tailed Recognition* (MDLT) as learning from multi-domain imbalanced data, with each domain having its own imbalanced label distribution, and generalizing to a test set that is balanced over all domain-class pairs. MDLT is a natural extension of the single domain case. It arises in real-world scenarios, where data targeted for one task can originate from different domains. For example, in visual recognition problems, minority classes from "photo" images could be complemented with potentially abundant samples from "sketch" images. Similarly, in autonomous

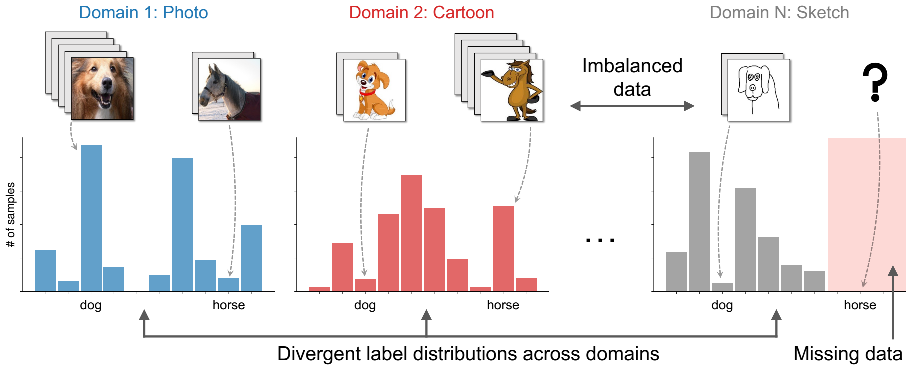


Fig. 1. Multi-Domain Long-Tailed Recognition (MDLT) aims to learn from imbalanced data from multiple distinct domains, tackle label imbalance, domain shift, and divergent label distributions across domains, and generalize to all domain-class pairs.

driving, the minority accident class in "real" life could be enriched with accidents generated in "simulation". Also, in medical diagnosis, data from distinct populations could enhance each other, where minority samples from one institution could be enriched with instances from others. In the above examples, different data types act as distinct *domains*, and such multi-domain data could be leveraged to tackle the inherent data imbalance within each domain.

We note that MDLT has key differences from its single-domain counterpart:

- First, the label distribution for each domain is likely different from other domains. For example, in Fig. 1, both "Photo" and "Cartoon" domains exhibit imbalanced label distributions; Yet, the "horse" class in "Cartoon" has many more samples than in "Photo". This creates challenges with *divergent label distributions across domains*, in addition to in-domain data imbalance.
- Second, multi-domain data inherently involves *domain shift*. Simply treating different domains as a whole and applying traditional data-imbalance methods is unlikely to yield the best results, as the domain gap can be arbitrarily large.
- Third, MDLT naturally motivates *zero-shot generalization within and across domains* – i.e., to generalize to both in-domain missing classes (Fig. 1 right part), as well as new domains with no training data, where the latter case is typically denoted as Domain Generalization (DG).

To deal with the above issues, we first develop the *domain-class transferability graph*, which quantifies the transferability between different domain-class pairs under data imbalance. In this graph, each node refers to a domain-class pair, and each edge refers to the distance between two domain-class pairs in the embedding space. We show that the transferability graph dictates the performance of imbalanced learning across domains. Inspired by this, we design `BoDA` (Balanced Domain-Class Distribution Alignment), a new loss function that encourages similarity between features of the same class in different domains, and penalizes similarity between features of different classes within and across domains. `BoDA` does so while accounting for that different classes have very different number of samples, and hence the statistics of their features are intrinsically imbalanced.

Analytically, we prove that minimizing the BoDA loss optimizes an upper bound of the *balanced* transferability statistics, corroborating the effectiveness of BoDA for learning multi-domain imbalanced data.

For MDLT evaluation, we curate five MDLT benchmarks based on datasets widely used for domain generalization (DG). These datasets naturally exhibit heavy class imbalance within each domain and data shift across domains, highlighting that the MDLT problem is widely present in current benchmarks. We compare BoDA against twenty algorithms that span different learning strategies. Extensive experiments across benchmarks and algorithms verify that BoDA consistently outperforms all these baselines on all datasets.

Additionally, we examine how BoDA performs in the DG setting. We show that combining BoDA with the DG state-of-the-art (SOTA) consistently brings further gains, yielding a new SOTA for DG. These results shed light on how label imbalance can affect out-of-distribution generalization and highlight the importance of integrating label imbalance into practical DG algorithm design.

Our contributions are as follows:

- We formulate the MDLT problem as learning from multi-domain imbalanced data and generalizing across all domain-class pairs.
- We introduce the domain-class transferability graph, a unified model for investigating MDLT. We further show that the transferability statistics induced from such graph are crucial and govern the success of MDLT algorithms.
- We design BoDA, a simple, effective, and interpretable loss function for MDLT. We prove theoretically that minimizing the BoDA loss is equivalent to optimizing an upper bound of balanced transferability statistics.
- Extensive experiments on benchmark datasets verify the superior and consistent performance of BoDA. Further, combined with DG algorithms, BoDA establishes a new SOTA on DG benchmarks, highlighting the importance of tackling cross-domain data imbalance for domain generalization.

2 Related Work

Long-Tailed Recognition. The literature is rich with research on long-tailed recognition [33, 57]. Proposed solutions include re-balancing the data by over-sampling/under-sampling [9, 20], re-weighting or adjusting the loss functions [6, 10, 12, 22], as well as leveraging relevant learning paradigms such as transfer learning [33], metric learning [55], meta-learning [43], two-stage training [23], ensemble learning [48, 56], and self-supervised learning [30, 52]. Recent studies have also explored imbalanced regression [53]. In contrast to these past works, we extend long-tailed recognition to the multi-domain setting, and introduce new techniques suitable for learning from multi-domain imbalanced data.

Multi-Domain Learning. Multi-domain learning (MDL) aims to learn a model of minimal risk from datasets drawn from different underlying distributions [13], and is a specific case of transfer learning [37]. In contrast to domain adaptation (DA) [3,37], which aims to minimize the risk over a single "target" domain, MDL minimizes the risk over all "source" domains, and considers both average and worst risks over all distributions [41]. Past solutions for MDL include designing shared and domain-specific models [13, 49], leveraging multi-task learning [51], and learning domain-invariant features [15, 31, 41, 45]. Our work falls under the MDL framework, but considers the practical and realistic setting where the label distribution is imbalanced within each domain and across domains.

Domain Generalization. Unlike MDL which focuses on in-domain generalization, domain generalization (DG) aims to learn from multiple training domains and generalize to unseen domains [59]. Previous approaches include learning domain-invariant features [15, 31, 34], learning transferable model parameters using meta-learning [27, 54], data augmentation [7, 60], and capturing causal relationships [1, 25]. Past work on DG has not investigated label imbalance within a domain and across domains. This paper shows that label imbalance plays a crucial role in DG, and that by combating data imbalance, we substantially boost DG performance on standard benchmarks.

3 Domain-Class Transferability Graph

When learning from MDLT, a natural question arises: How do we model MDLT in the presence of both *domain shift* and *class imbalance within and across domains*? We argue that in contrast to single-domain imbalanced learning where the basic unit one cares about is a *class* (i.e., minority *vs.* majority classes), in MDLT, the basic unit naturally translates to a **domain-class pair**.

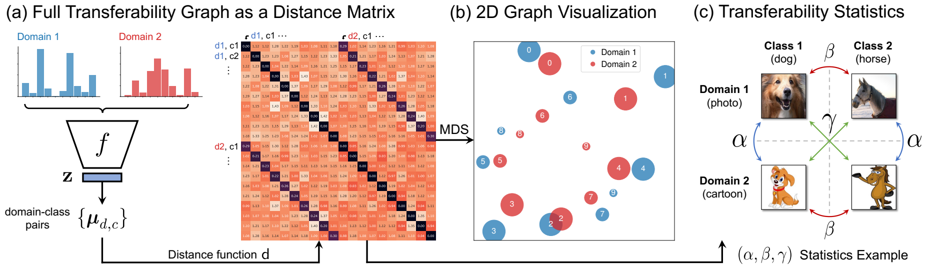
Problem Setup. Given a multi-domain classification task with a discrete label space $\mathcal{C} = \{1, \ldots, C\}$ and a domain space $\mathcal{D} = \{1, \ldots, D\}$, let $\mathcal{S} = \{(\mathbf{x}_i, c_i, d_i)\}_{i=1}^{N}$ be the training set, where $\mathbf{x}_i \in \mathbb{R}^l$ denotes the input, $c_i \in \mathcal{C}$ is the class label, and $d_i \in \mathcal{D}$ is the domain label. We denote as $\mathbf{z} = f(\mathbf{x}; \theta)$ the representation of $\mathbf{x}$, where $f : \mathcal{X} \to \mathcal{Z}$ maps the input into a representation space $\mathcal{Z} \subseteq \mathbb{R}^h$. The final prediction $\widehat{c} = g(\mathbf{z})$ is given by a classification function $g : \mathcal{Z} \to \mathcal{C}$. We denote the set of samples belonging to domain d and class c (i.e., the domain-class pair (d, c)) as $\mathcal{S}_{d,c} \subseteq \mathcal{S}$, with $N_{d,c} \triangleq |\mathcal{S}_{d,c}|$ as the number of samples. Similarly, $\mathcal{Z}_{d,c} \subseteq \mathcal{Z}$ denotes the representation set for (d, c). We use $\mathcal{M} = \mathcal{D} \times \mathcal{C} := \{(d, c) : d \in \mathcal{D}, c \in \mathcal{C}\}$ to denote the set of all domain-class pairs.

Definition 1 (Transferability). *Given a learned model and a distance function* $\mathsf{d} : \mathbb{R}^h \times \mathbb{R}^h \to \mathbb{R}$ *in the feature space, the transferability from domain-class pair* (d, c) *to* (d', c') *is:*

$$\operatorname{trans}\big((d, c), (d', c')\big) \triangleq \mathbb{E}_{\mathbf{z} \in \mathcal{Z}_{d,c}} \left[\mathsf{d}\left(\mathbf{z}, \boldsymbol{\mu}_{d',c'}\right) \right],$$

where $\boldsymbol{\mu}_{d',c'} \triangleq \mathbb{E}_{\mathbf{z}' \in \mathcal{Z}_{d',c'}}[\mathbf{z}']$ *is the first order statistics (i.e., mean) of* (d', c').

Intuitively, the transferability between two domain-class pairs is the average distance between their learned representations, characterizing how close they are in the feature space. By default, d is chosen as the Euclidean distance, but it can also represent the higher order statistics of (d, c). For example, the Mahalanobis distance [11] uses the covariance $\boldsymbol{\Sigma}_{d,c} \triangleq \mathbb{E}_{\mathbf{z} \in \mathcal{Z}_{d,c}} \left[(\mathbf{z} - \boldsymbol{\mu}_{d,c})(\mathbf{z} - \boldsymbol{\mu}_{d,c})^\top \right]$. In the remainder of the paper, with a slight abuse of the notation, we allow $\boldsymbol{\mu}_{d,c}$ to represent both the first and higher order statistics for (d, c).



(a) Full Transferability Graph as a Distance Matrix

(b) 2D Graph Visualization

(c) Transferability Statistics

Fig. 2. Overall framework of transferability graph. **(a)** Distribution statistics $\{\boldsymbol{\mu}_{d,c}\}$ is computed for all domain-class pairs, by which we generate a full transferability matrix. **(b)** MDS is used to project the graph into a 2D space for visualization. **(c)** We define (α, β, γ) transferability statistics to further describe the whole transferability graph.

Definition 2 (Transferability Graph). *The transferability graph for a learned model is defined as* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where the vertices,* $\mathcal{V} \subseteq \{\boldsymbol{\mu}_{d,c}\}$*, represents the domain-class pairs, and the edges,* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$*, are assigned weights equal to* $\mathrm{trans}\left((d, c), (d', c')\right)$.

Transferability Graph Visualization. It is convenient to visualize the transferability graph of a learned model in a 2D Cartesian space. To do so, we use the average of $\mathrm{trans}\left((d, c), (d', c')\right)$ and $\mathrm{trans}\left((d', c'), (d, c)\right)$ as a similarity measure between them. We can then visualize this similarity and the underlying transferability graph using multidimensional scaling (MDS) [8]. Figs. 2a and 2b show this process, where for each (d, c) pair, we estimate its distribution statistics $\{\boldsymbol{\mu}_{d,c}\}$ from the learned model, then compute the model transferability graph as a distance matrix. We then use MDS to project it into a 2D space, where each dot refers to one (d, c), and the distance represents transferability.

Definition 3 ((α, β, γ) Transferability Statistics). *The transferability graph can be summarized by the following transferability statistics:*

$$\begin{aligned}
\textit{Different domains, same class:} \quad & \alpha = \mathbb{E}_c \mathbb{E}_d \mathbb{E}_{d' \neq d} \left[\mathrm{trans}\big((d, c), (d', c)\big)\right] . \\
\textit{Same domain, different classes:} \quad & \beta = \mathbb{E}_d \mathbb{E}_c \mathbb{E}_{c' \neq c} \left[\mathrm{trans}\big((d, c), (d, c')\big)\right] . \\
\textit{Different domains, different classes:} \quad & \gamma = \mathbb{E}_d \mathbb{E}_{d' \neq d} \mathbb{E}_c \mathbb{E}_{c' \neq c} \left[\mathrm{trans}\big((d, c), (d', c')\big)\right] .
\end{aligned}$$

As illustrated in Fig. 2c, (α, β, γ) captures the similarity between features of the same class across domains and different classes within and across domains.

4 What Makes for Good Representations in MDLT?

4.1 Divergent Label Distributions Hamper Transferable Features

MDLT has to deal with differences between the label distributions across domains. To understand the implications of this issue we start with an example.

Motivating Example. We construct `Digits-MLT`, a two-domain toy MDLT dataset that combines two digit datasets: MNIST-M [15] and SVHN [36]. The task is 10-class digit classification. Details of the datasets are in Appendix D.

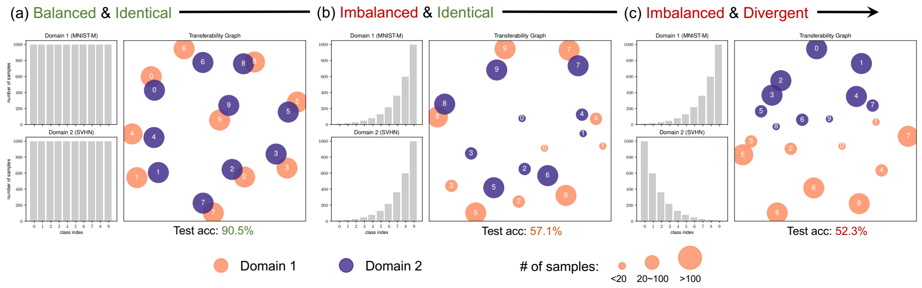


Fig. 3. The evolving pattern of transferability graph when varying label proportions of `Digits-MLT`. **(a)** Label distribution for two domains are balanced and identical. **(b)** Label distribution for two domains are imbalanced but identical. **(c)** Label distribution for two domains are imbalanced and *divergent*.

We manually vary the number of samples for each domain-class pair to simulate different label distributions, and train a plain ResNet-18 [21] using empirical risk minimization (ERM) for each case. We keep all test sets balanced and identical.

The results in Fig. 3 reveal interesting observations. When the per-domain label distributions are balanced and *identical* across domains, although a domain gap exists, it does not prohibit the model from learning discriminative features of high accuracy (90.5%), as shown in Fig. 3a. If the label distributions are imbalanced but *identical*, as in Fig. 3b, ERM is still able to align similar classes in the two domains, where majority classes (e.g., class 9) are closer in terms of transferability than minority classes (e.g., class 0). In contrast, when the labels are both imbalanced and *mismatched* across domains, as in Fig. 3c, the learned features are no longer transferable, resulting in a clear gap across domains and the worst accuracy. This is because *divergent label distributions* across domains produce an undesirable shortcut; the model can minimize the classification loss simply by separating the two domains.

Transferable Features are Desirable. As the results indicate, *transferable* features across (d, c) pairs are needed, especially when imbalance occurs. In particular, the transferability link between the same class across domains should be greater than that between different classes within or across domains. This can be captured via the (α, β, γ) transferability statistics, as we show next.

4.2 Transferability Statistics Characterize Generalization

Motivating Example. Again, we use `Digits-MLT` with varying label distributions. We consider three imbalance types to compose different label configurations: (1) **Uniform** (i.e., balanced labels), (2) **Forward-LT**, where the labels exhibit a long tail over class ids, and (3) **Backward-LT**, where labels are inversely long-tailed with respect to the class ids. For each configuration, we train 20 ERM models with varying hyperparameters. We then calculate the (α, β, γ) statistics for each model, and plot its classification accuracy against $(\beta+\gamma)-\alpha$.

Fig. 4 reveals the following findings: (1) *The* (α, β, γ) *statistics characterize a model's performance in MDLT.* In particular, the $(\beta+\gamma)-\alpha$ quantity displays a very strong correlation with test performance across the entire range and

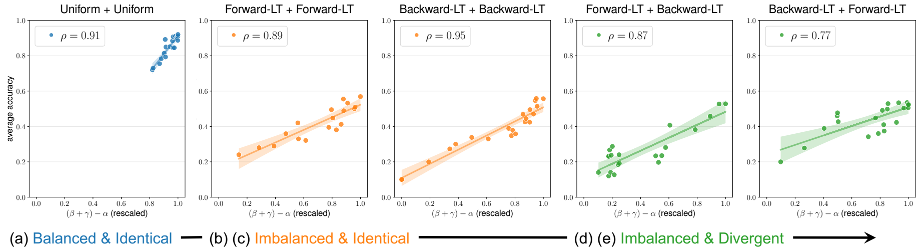


Fig. 4. Correspondence between $(\beta+\gamma)-\alpha$ quantity and test accuracy across different label configurations of `Digits-MLT`. Each plot refers to specific label distributions for two domains (e.g., (a) employs "Uniform" for domain 1 and "Uniform" for domain 2). Each point corresponds to a model trained with ERM using different hyperparameters.

every label configuration. (2) *Data imbalance increases the risk of learning less transferable features.* When the label distributions are similar across domains (Fig. 4a), the models are robust to varying parameters, clustering in the upper-right region. However, as the labels become imbalanced (Figs. 4b, 4c) and further divergent (Figs. 4d, 4e), chances that the model learns non-transferable features (i.e., lower $(\beta+\gamma)-\alpha$) increase, leading to a large drop in performance. We provide further evidence in Appendix H.4 showing that these observations hold regardless of datasets and training regimes.

4.3 A Loss that Bounds the Transferability Statistics

We use the above findings to design a new loss function particularly suitable for MDLT. We will first introduce the loss function then prove that it minimizes an upper bound of the (α,β,γ) statistics. We start from a simple loss inspired by the metric learning objective [17, 44]. We call this loss $\mathcal{L}_{\texttt{DA}}$ since it aims for Domain Alignment, i.e., aligning the features of the same class across domains. Let $(\mathbf{x}_i, c_i, d_i)$ denote a sample with feature $\mathbf{z}_i$. Given a set of training samples with feature set $\mathcal{Z}$, we have

$$\mathcal{L}_{\texttt{DA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) = \textstyle\sum_{\mathbf{z}_i\in\mathcal{Z}} \frac{-1}{|\mathcal{D}|-1}\sum_{d\in\mathcal{D}\backslash\{d_i\}} \log \frac{\exp\left(-\mathsf{d}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\backslash\{(d_i,c_i)\}}\exp\left(-\mathsf{d}(\mathbf{z}_i,\boldsymbol{\mu}_{d',c'})\right)}. \quad (1)$$

Intuitively, $\mathcal{L}_{\texttt{DA}}$ tackles label *divergence*, as (d,c) pairs that share same class would be pulled closer, and vice versa. It is also related to (α,β,γ) because the numerator represents *positive* cross-domain pairs (α), and the denominator represents *negative* cross-class pairs (β,γ). A detailed probabilistic interpretation of $\mathcal{L}_{\texttt{DA}}$ is provided in Appendix B.2.

But, $\mathcal{L}_{\texttt{DA}}$ does not address label *imbalance*. Note that (α,β,γ) is defined in a *balanced* way, independent of the number of samples of each (d,c). However, given an imbalanced dataset, most samples will come from majority domain-class pairs, which would dominate $\mathcal{L}_{\texttt{DA}}$ and cause minority pairs to be overlooked.

Balanced Domain-Class Distribution Alignment (BoDA). To tackle data imbalance across (d,c) pairs, we modify the loss in Eqn. (1) to the `BoDA` loss:

$$\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) = \sum_{\mathbf{z}_i\in\mathcal{Z}} \frac{-1}{|\mathcal{D}|-1}\sum_{d\in\mathcal{D}\backslash\{d_i\}} \log \frac{\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\backslash\{(d_i,c_i)\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i,\boldsymbol{\mu}_{d',c'})\right)}, \quad \widetilde{\mathsf{d}}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c}) = \frac{\mathsf{d}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c})}{N_{d_i,c_i}}. \quad (2)$$

$\texttt{BoDA}$ scales the original d by a factor of $1/N_{d_i,c_i}$, i.e., it counters the effect of imbalanced domain-class pairs by introducing a *balanced* distance measure $\widetilde{\mathsf{d}}$.

Theorem 1 ($\mathcal{L}_{\texttt{BoDA}}$ as an Upper Bound). *Given a multi-domain long-tailed dataset $\mathcal{S}$ with domain label space $\mathcal{D}$ and class label space $\mathcal{C}$ satisfying $|\mathcal{D}| > 1$ and $|\mathcal{C}| > 1$, let $\mathcal{Z}$ be the representation set of all training samples, and (α, β, γ) be the transferability statistics for $\mathcal{S}$ defined in Definition 3. It holds that*

$$\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) \geq N \log\left(|\mathcal{D}| - 1 + |\mathcal{D}|(|\mathcal{C}| - 1)\exp\left(\frac{|\mathcal{C}||\mathcal{D}|}{N} \cdot \alpha - \frac{|\mathcal{C}|}{N} \cdot \beta - \frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N} \cdot \gamma\right)\right). \quad (3)$$

The proof of Theorem 1 is in Appendix A.2. Theorem 1 has the following interesting implications: (1) $\mathcal{L}_{\texttt{BoDA}}$ *upper-bounds* (α, β, γ) *statistics in a desired form that naturally translates to better performance.* By minimizing $\mathcal{L}_{\texttt{BoDA}}$, we ensure a low α (attract same classes) and high β, γ (separate different classes), which are essential conditions for generalization in MDLT. (2) *The constant factors correspond to how much each component contributes to the transferability graph.* Zooming on the arguments of $\exp(\cdot)$, we observe that the objective is proportional to $\alpha - (\frac{1}{|\mathcal{D}|}\beta + \frac{|\mathcal{D}|-1}{|\mathcal{D}|}\gamma)$. According to Definition 3, we note that α summarizes data similarity for the same class, while $(\frac{1}{|\mathcal{D}|}\beta + \frac{|\mathcal{D}|-1}{|\mathcal{D}|}\gamma)$ summarizes data similarity across different classes, using the weighted average of β and γ, where their weights are proportional to the number of associated domains (i.e., 1 for β, $(|\mathcal{D}| - 1)$ for γ).

4.4 Calibration for Data Imbalance Leads to Better Transfer

$\texttt{BoDA}$ works by encouraging feature transfer for similar classes across domains, i.e., if (d, c) and (d', c) refer to the same class in different domains, then we want to transfer their features to each other. But, minority domain-class pairs naturally have worse $\boldsymbol{\mu}_{d,c}$ estimates due to data scarcity, and forcing other pairs to transfer to them hurts learning. Thus, when bringing two domain-class pairs closer in the embedding space, we want the minority (d, c) to transfer to majority ones, not the inverse. The following example further clarifies this point.

Motivating Example. We use Digits-MLT with divergent labels (Fig. 5). We focus on *feature discrepancy*, i.e., the distance between training and test features for the same class. For each class in domain 1, we compute the distance in the feature space between the means of the training set and test set (solid line). We also compute the distance between the training data of domain 2 and test data of domain 1 (dashed line), for the same class.

As shown by the solid orange line in Fig. 5b, for minority domain-class pairs such as class "8" and "9" in domain 1, the distance in the feature space between training and testing is large. In fact, the test set of these minority domain-class pairs is closer to the training data for "8" and "9" in domain 2 than in their own domain, as shown by the dashed purple line. This example indicates that a better training would try to transfer the features of minority domain-class pairs to majority pairs with which they share the same class, as shown by the grey arrow in Fig. 5b. Such transfer will improve generalization to the test set.

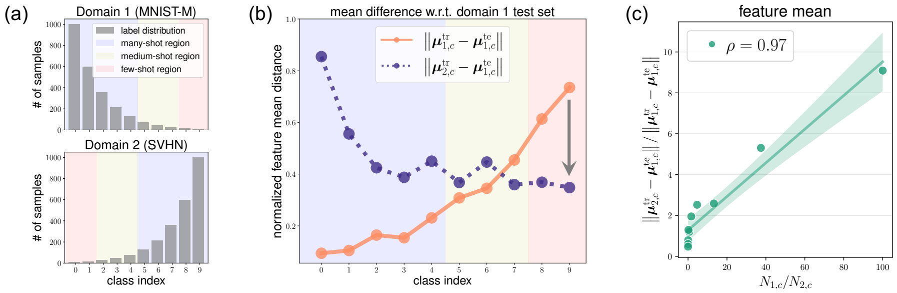


Fig. 5. The need for *calibration.* **(a)** Per-domain label distribution. **(b)** Distance between training and test data. Solid plots the distance between training and test data from the same domain-class pairs. Dashed plots the distance between test data from a particular domain-class pair and the training data with which it shares the same class but differs in the domain. The blue and red background colors refer to majority and minority domain-class pairs, respectively. **(c)** Correspondence between the feature distance ratio and the sample size ratio for two domain-class pairs.

`BoDA` **with Calibrated Distance.** The above discussion motivates a modification to `BoDA` to favor transfer to majority domain-class pairs:

$$\widetilde{\mathcal{L}}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) = \sum_{\mathbf{z}_i \in \mathcal{Z}} \frac{-1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\lambda^{d,c_i}_{d_i,c_i} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c') \in \mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\lambda^{d',c'}_{d_i,c_i} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}, \quad \lambda^{d',c'}_{d,c} = \left(\frac{N_{d',c'}}{N_{d,c}}\right)^{\nu}, \quad (4)$$

where ν is a constant that allows for a sublinear relation (default $\nu = 1$). $\lambda^{d',c'}_{d,c}$ indicates how much we would like to transfer (d, c) to (d', c'), based on their relative sample size. Fig. 5c verifies that the ratio of the sample size is highly correlated with the ratio of the distance between testing and training. Further, Theorem 2 in Appendix A shows that $\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ is an upper bound of the calibrated transferability statistics.

Variants of `BoDA`: Matching Higher Order Statistics. The distance d can be set to the Euclidean distance $\mathsf{d}(\mathbf{z}, \boldsymbol{\mu}_{d,c}) = \sqrt{(\mathbf{z} - \boldsymbol{\mu}_{d,c})^\top (\mathbf{z} - \boldsymbol{\mu}_{d,c})}$, which captures the first order statistics. To match higher order statistics such as covariance, we set $\mathsf{d}(\mathbf{z}, \{\boldsymbol{\mu}_{d,c}, \boldsymbol{\Sigma}_{d,c}\}) = \sqrt{(\mathbf{z} - \boldsymbol{\mu}_{d,c})^\top \boldsymbol{\Sigma}^{-1}_{d,c} (\mathbf{z} - \boldsymbol{\mu}_{d,c})}$, resembling the Mahalanobis distance [11]. We refer to these variants as $\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ and $\widetilde{\mathcal{L}}_{\texttt{BoDA-M}}$.

Joint Loss. `BoDA` serves as a representation learning scheme for MDLT, which operates over $\mathcal{Z}$. For classification, we train deep networks by combining $\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ and the standard cross-entropy (CE) loss in an end-to-end fashion, where CE is applied to the output layer, and `BoDA` is applied to the latent features. We combine the losses as $\mathcal{L}_{\texttt{CE}} + \omega \widetilde{\mathcal{L}}_{\texttt{BoDA}}$, with ω as a trade-off hyperparameter.

5 What Makes for Good Classifiers in MDLT?

In the long-tailed recognition literature, an important finding is that decoupling *representation learning* and *classifier learning* leads to better results [23, 58]. In particular, instance-balanced sampling is used during the first stage of learning, while class-balanced sampling is used for re-training the classifier (with the representation fixed) in the second stage [23]. Motivated by this, we explore whether a similar decoupling benefits MDLT. We use three learning algorithms, ERM [46], DANN [31], and CORAL [45]. We train each algorithm with and without the second stage classifier learning, and report the average accuracy over all MDLT datasets (presented later).

As Table 1 shows, similar to what has been observed in the single domain case [23,58], regardless of algorithm, decoupling the classifier learning consistently improves performance. Since BoDA can support both coupled and decoupled classifier learning, we use BoDA$_r$ to refer to models that couple representation and classifier learning, and BoDA$_{r,c}$ for models that decouple representation from classifier learning. In the classifier learning stage, we simply use class-balanced sampling.

Table 1. The benefits of decoupling the classifier.

Algorithm	w/o decouple	w/ decouple
ERM [46]	77.6 ±0.2	**79.2** ±0.3
DANN [15]	77.7 ±0.6	**79.0** ±0.1
CORAL [45]	78.0 ±0.1	**79.6** ±0.2

6 Benchmarking MDLT

Datasets. We curate five multi-domain datasets typically used in DG and adapt them for MDLT evaluation. To do so, for each dataset, we create two balanced datasets one for validation and the other for testing, and leave the rest for training. The size of the validation and test data sets is 5% and 10% of original data, respectively. Table 10 in Appendix D provides the statistics of each MDLT dataset. Fig. 6 shows the label distributions across domains in the five datasets.

1. VLCS-MLT. We construct VLCS-MLT using the VLCS dataset [14], which is an object recognition dataset with 10,729 images from 4 domains and 5 classes.
2. PACS-MLT. PACS-MLT is constructed from the PACS dataset [28], an object recognition dataset with 9,991 images from 4 domains and 7 classes.
3. OfficeHome-MLT. We set up OfficeHome-MLT using the OfficeHome dataset [47] which contains 15,588 images from 4 domains and 65 classes.
4. TerraInc-MLT. TerraInc-MLT is created from TerraIncognita [2], a species classification dataset including 24,788 images from 4 domains and 10 classes.
5. DomainNet-MLT. We construct DomainNet-MLT using DomainNet [38], a large-scale multi-domain dataset for object recognition. It contains 586,575 images from 345 classes and 6 domains.

Network Architectures. For experiments on the synthetic Digits-MLT dataset, we use a simple CNN architecture as in [19]. For the MDLT datasets, we follow [19], and use ResNet-50 [21] for all algorithms.

Competing Algorithms. We compare BoDA to a large number of algorithms that span different learning strategies and categories, including (1) *vanilla:* **ERM** [46], (2) *distributionally robust optimization:* **GroupDRO** [40], (3) *data augmentation:* **Mixup** [50], **SagNet** [35], (4) *meta-learning:* **MLDG** [27], (5) *domain-invariant feature learning:* **IRM** [1], **DANN** [15], **CDANN** [31], **CORAL** [45], **MMD** [29], (6) *transfer learning:* **MTL** [4], (7) *multi-task learning:* **Fish** [42], and (8) *imbalanced learning:* **Focal** [32], **CBLoss** [10], **LDAM** [6], **BSoftmax** [39], **SSP** [52], **CRT** [23]. We provide detailed descriptions in Appendix E.

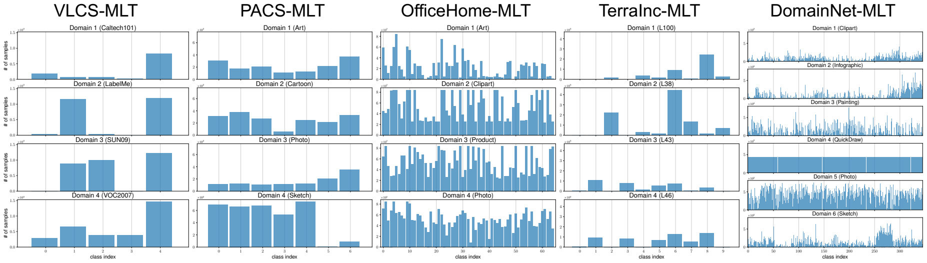


Fig. 6. Overview of training set label distribution for five MDLT datasets. We set up MDLT benchmarks from datasets traditionally used for DG, and make validation/test sets balanced across all domain-class pairs. More details are provided in Appendix D.

Table 2. Results on `VLCS-MLT`.

	Accuracy (by domain)		Accuracy (by shot)			
Algorithm	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM [46]	76.3 ±0.4	53.6 ±1.1	84.6 ±0.5	76.6 ±0.4	–	32.9 ±0.4
IRM [1]	76.5 ±0.2	52.3 ±0.7	85.3 ±0.6	75.5 ±1.0	–	33.5 ±1.0
GroupDRO [40]	76.7 ±0.4	54.1 ±1.3	85.3 ±0.9	76.2 ±1.0	–	34.5 ±2.0
Mixup [50]	75.9 ±0.1	52.7 ±1.3	84.4 ±0.2	77.1 ±0.6	–	29.2 ±1.4
MLDG [27]	76.9 ±0.2	53.6 ±0.5	84.9 ±0.3	77.5 ±1.0	–	34.4 ±0.9
CORAL [45]	75.9 ±0.5	51.6 ±0.7	84.3 ±0.6	75.5 ±0.5	–	34.5 ±0.8
MMD [29]	76.3 ±0.6	53.4 ±0.3	84.5 ±0.8	77.1 ±0.5	–	32.7 ±0.3
DANN [15]	77.5 ±0.1	54.1 ±0.3	85.9 ±0.5	76.0 ±0.4	–	38.0 ±2.3
CDANN [31]	76.6 ±0.4	53.6 ±0.4	84.4 ±0.7	77.3 ±0.8	–	35.0 ±0.8
MTL [4]	76.3 ±0.3	52.9 ±0.5	84.8 ±0.9	76.2 ±0.6	–	33.3 ±1.4
SagNet [35]	76.3 ±0.2	52.3 ±0.2	85.3 ±0.3	75.1 ±0.2	–	32.9 ±0.3
Fish [42]	77.5 ±0.3	54.3 ±0.4	86.2 ±0.5	76.0 ±0.4	–	35.6 ±2.2
Focal [32]	75.6 ±0.4	52.3 ±0.2	84.0 ±0.2	75.5 ±0.6	–	32.7 ±0.9
CBLoss [10]	76.8 ±0.3	52.5 ±0.5	84.8 ±0.7	77.5 ±1.4	–	33.2 ±1.6
LDAM [6]	77.5 ±0.1	52.9 ±0.2	**86.5** ±0.4	75.5 ±0.5	–	35.2 ±0.6
BSoftmax [39]	76.7 ±0.5	52.9 ±0.9	84.4 ±0.9	78.2 ±0.6	–	34.3 ±0.9
SSP [52]	76.1 ±0.3	52.3 ±1.0	83.8 ±0.3	76.0 ±1.2	–	37.1 ±0.7
CRT [23]	76.3 ±0.2	51.4 ±0.3	84.5 ±0.1	77.3 ±0.0	–	31.7 ±1.0
BoDA$_r$	76.9 ±0.5	51.4 ±0.3	85.3 ±0.3	77.3 ±0.2	–	33.3 ±0.5
BoDA-M$_r$	77.5 ±0.3	53.4 ±0.3	85.8 ±0.2	77.3 ±0.2	–	35.7 ±0.7
BoDA$_{r,c}$	77.3 ±0.2	53.4 ±0.3	85.3 ±0.3	78.0 ±0.2	–	38.6 ±0.7
BoDA-M$_{r,c}$	**78.2** ±0.4	**55.4** ±0.5	85.3 ±0.3	**79.3** ±0.6	–	**43.3** ±1.1
BoDA *vs.* ERM	**+1.9**	**+1.8**	**+0.7**	**+2.7**	–	**+10.4**

Table 3. Results on `PACS-MLT`.

	Accuracy (by domain)		Accuracy (by shot)			
Algorithm	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM [46]	97.1 ±0.1	95.8 ±0.2	97.1 ±0.0	97.0 ±0.0	98.0 ±0.9	–
IRM [1]	96.7 ±0.2	95.2 ±0.4	96.8 ±0.2	96.7 ±0.7	94.7 ±1.4	–
GroupDRO [40]	97.0 ±0.1	95.3 ±0.4	97.3 ±0.1	95.3 ±1.2	94.7 ±3.6	–
Mixup [50]	96.7 ±0.2	95.1 ±0.2	97.0 ±0.1	96.7 ±0.3	91.3 ±2.7	–
MLDG [27]	96.6 ±0.1	94.1 ±0.3	96.8 ±0.1	96.3 ±0.7	92.7 ±0.5	–
CORAL [45]	96.6 ±0.5	94.3 ±0.7	96.6 ±0.5	97.0 ±0.8	94.7 ±0.5	–
MMD [29]	96.9 ±0.1	96.2 ±0.2	96.9 ±0.2	97.0 ±0.0	96.7 ±0.5	–
DANN [15]	96.5 ±0.0	94.3 ±0.1	96.5 ±0.1	98.0 ±0.0	94.7 ±2.4	–
CDANN [31]	96.1 ±0.1	94.5 ±0.2	96.1 ±0.1	96.3 ±0.5	94.0 ±0.9	–
MTL [4]	96.7 ±0.2	94.5 ±0.6	96.8 ±0.1	95.3 ±1.7	97.3 ±1.1	–
SagNet [35]	**97.2** ±0.1	95.2 ±0.3	**97.4** ±0.1	96.7 ±0.5	95.3 ±0.5	–
Fish [42]	96.9 ±0.2	95.2 ±0.2	97.0 ±0.1	97.0 ±0.5	94.7 ±1.1	–
Focal [32]	96.5 ±0.2	94.6 ±0.7	96.6 ±0.1	95.0 ±1.7	96.7 ±0.5	–
CBLoss [10]	96.9 ±0.1	95.1 ±0.4	96.8 ±0.2	97.0 ±1.2	**100.0** ±0.0	–
LDAM [6]	96.5 ±0.2	94.7 ±0.2	96.6 ±0.1	95.7 ±1.4	96.0 ±0.0	–
BSoftmax [39]	96.9 ±0.3	95.6 ±0.3	96.6 ±0.4	**98.7** ±0.7	99.3 ±0.5	–
SSP [52]	96.9 ±0.2	95.4 ±0.4	96.7 ±0.2	98.3 ±0.5	98.0 ±0.9	–
CRT [23]	96.3 ±0.1	94.9 ±0.1	96.3 ±0.1	97.3 ±0.3	94.0 ±0.9	–
BoDA$_r$	97.0 ±0.1	95.1 ±0.4	97.0 ±0.1	96.3 ±0.5	98.0 ±0.9	–
BoDA-M$_r$	97.1 ±0.1	94.9 ±0.1	97.3 ±0.1	96.3 ±0.5	96.0 ±0.0	–
BoDA$_{r,c}$	**97.2** ±0.1	95.7 ±0.3	**97.4** ±0.1	97.0 ±0.0	94.7 ±1.1	–
BoDA-M$_{r,c}$	97.1 ±0.2	**96.3** ±0.1	97.1 ±0.0	97.0 ±0.8	96.0 ±0.0	–
BoDA *vs.* ERM	**+0.1**	**+0.5**	**+0.3**	**+0.0**	**-2.0**	–

Implementation and Evaluation Metrics. For a fair evaluation, following [19], for each algorithm we conduct a random search of 20 trials over a joint distribution of all hyperparameters (see Appendix E.3 for details). We then use the validation set to select the best hyperparameters for each algorithm, fix them and rerun the experiments under three different random seeds to report the final average accuracy with standard deviation. Such process ensures the comparison is best-versus-best, and the hyperparameters are optimized for all algorithms. In addition to the average accuracy across domains, we also report the worst accuracy over domains, and further divide all domain-class pairs into *many-shot* (pairs with over 100 training samples), *medium-shot* (pairs with 20∼100 training samples), *few-shot* (pairs with under 20 training samples), and *zero-shot* (pairs with no training data), and report the results for these subsets.

6.1 Main Results

We report the main results in this section for all MDLT datasets. The complete results and all additional experiments are provided in Appendix F and H.

Benchmark Results on MDLT Datasets. The performance of all methods on `VLCS-MLT`, `PACS-MLT`, `OfficeHome-MLT`, `TerraInc-MLT` and `DomainNet-MLT`

Table 4. Results on `OfficeHome-MLT`.

	Accuracy (by domain)		Accuracy (by shot)			
Algorithm	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM [46]	80.7 ±0.0	71.3 ±0.1	87.8 ±0.2	81.0 ±0.2	63.1 ±0.1	63.3 ±7.2
IRM [1]	80.6 ±0.4	70.7 ±0.2	87.6 ±0.4	81.5 ±0.4	61.1 ±0.9	56.7 ±1.4
GroupDRO [40]	80.1 ±0.3	68.7 ±0.9	88.1 ±0.2	80.8 ±0.4	59.8 ±1.2	51.7 ±3.6
Mixup [50]	81.2 ±0.2	72.3 ±0.6	87.9 ±0.4	81.8 ±0.1	64.1 ±0.4	60.0 ±4.1
MLDG [27]	80.4 ±0.2	70.2 ±0.6	87.1 ±0.1	81.3 ±0.3	61.3 ±1.0	61.7 ±1.4
CORAL [45]	81.9 ±0.1	**72.7** ±0.6	87.9 ±0.1	83.0 ±0.1	63.5 ±0.7	65.0 ±2.4
MMD [29]	78.4 ±0.4	67.7 ±0.8	85.2 ±0.2	79.4 ±0.7	58.8 ±0.4	56.7 ±3.6
DANN [15]	79.2 ±0.2	70.2 ±0.9	86.2 ±0.1	80.0 ±0.1	60.3 ±1.1	61.7 ±5.9
CDANN [31]	79.0 ±0.2	69.4 ±0.3	86.4 ±0.6	79.8 ±0.1	58.9 ±0.8	50.0 ±4.7
MTL [4]	79.5 ±0.2	69.8 ±0.6	87.3 ±0.3	79.8 ±0.2	61.1 ±0.2	51.7 ±2.7
SagNet [35]	80.9 ±0.1	70.5 ±0.5	87.8 ±0.4	81.9 ±0.1	61.2 ±0.9	56.7 ±3.6
Fish [42]	81.3 ±0.3	71.3 ±0.7	**88.2** ±0.2	81.9 ±0.3	63.2 ±0.8	61.7 ±1.4
Focal [32]	77.9 ±0.0	67.6 ±0.4	86.5 ±0.3	78.3 ±0.1	57.4 ±0.3	46.7 ±3.6
CBLoss [10]	79.8 ±0.2	69.5 ±0.7	86.6 ±0.4	80.6 ±0.2	61.1 ±1.4	65.0 ±2.4
LDAM [6]	80.3 ±0.2	69.9 ±0.5	87.1 ±0.2	81.3 ±0.3	61.1 ±0.2	51.7 ±2.7
BSoftmax [39]	80.4 ±0.2	70.9 ±0.5	86.7 ±0.5	81.3 ±0.3	62.4 ±1.0	60.0 ±4.1
SSP [52]	81.1 ±0.3	71.1 ±0.3	87.3 ±0.6	82.3 ±0.3	61.6 ±0.7	63.3 ±1.4
CRT [23]	81.2 ±0.0	72.5 ±0.2	87.7 ±0.1	81.8 ±0.1	64.0 ±0.1	65.0 ±2.4
BoDA$_r$	81.5 ±0.1	71.8 ±0.1	87.7 ±0.2	82.3 ±0.1	**64.2** ±0.3	63.3 ±1.4
BoDA-M$_r$	81.9 ±0.2	71.6 ±0.2	87.3 ±0.3	83.4 ±0.2	62.3 ±0.3	65.0 ±2.4
BoDA$_{r,c}$	82.3 ±0.1	72.3 ±0.3	87.1 ±0.2	**83.9** ±0.3	63.2 ±0.2	65.0 ±2.4
BoDA-M$_{r,c}$	**82.4** ±0.2	72.3 ±0.3	87.7 ±0.1	**83.9** ±0.6	**64.2** ±0.3	**66.7** ±2.7
BoDA *vs.* ERM	**+1.7**	**+1.0**	**-0.1**	**+2.9**	**+1.1**	**+3.4**

Table 5. Results on `TerraInc-MLT`.

	Accuracy (by domain)		Accuracy (by shot)			
Algorithm	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM [46]	75.3 ±0.3	67.4 ±0.3	85.6 ±0.8	69.6 ±3.2	66.1 ±2.4	14.4 ±2.8
IRM [1]	73.3 ±0.7	64.3 ±1.3	83.5 ±0.6	70.0 ±1.8	58.3 ±3.4	20.1 ±1.4
GroupDRO [40]	72.0 ±0.4	66.6 ±0.2	84.7 ±1.1	64.6 ±4.7	38.9 ±1.2	13.5 ±1.1
Mixup [50]	71.1 ±0.7	60.4 ±1.1	83.2 ±0.7	60.0 ±0.6	56.1 ±3.0	12.2 ±2.1
MLDG [27]	76.6 ±0.2	66.9 ±0.5	86.1 ±0.6	73.8 ±3.9	70.6 ±3.7	18.8 ±2.4
CORAL [45]	76.4 ±0.5	67.8 ±0.9	86.3 ±0.3	77.5 ±3.1	66.1 ±2.0	11.0 ±1.4
MMD [29]	73.3 ±0.4	63.7 ±1.1	84.0 ±0.4	67.9 ±2.7	60.6 ±1.6	13.6 ±2.6
DANN [15]	68.7 ±0.9	61.1 ±1.0	79.6 ±1.2	62.5 ±8.1	48.9 ±2.8	13.3 ±1.1
CDANN [31]	70.3 ±0.5	63.9 ±1.0	83.5 ±0.8	50.0 ±4.2	43.9 ±4.7	20.4 ±3.1
MTL [4]	75.0 ±0.7	67.7 ±1.4	85.2 ±0.7	73.8 ±1.6	61.1 ±2.8	12.4 ±4.0
SagNet [35]	75.1 ±1.6	66.5 ±2.1	85.5 ±0.9	77.1 ±5.0	57.8 ±4.3	13.0 ±3.4
Fish [42]	75.3 ±0.5	66.3 ±0.5	85.8 ±0.2	73.3 ±3.9	61.1 ±3.0	13.7 ±3.3
Focal [32]	75.7 ±0.4	65.3 ±1.1	85.7 ±0.3	76.2 ±3.9	68.9 ±3.2	12.6 ±1.9
CBLoss [10]	78.0 ±0.4	68.3 ±2.0	85.0 ±0.1	89.2 ±1.2	83.9 ±2.5	9.3 ±3.9
LDAM [6]	74.7 ±0.9	64.1 ±1.4	85.1 ±0.6	70.8 ±3.5	67.8 ±1.2	11.1 ±2.4
BSoftmax [39]	76.7 ±1.0	65.6 ±1.3	83.4 ±0.8	90.8 ±0.9	78.3 ±3.9	12.6 ±2.4
SSP [52]	78.5 ±0.7	67.3 ±0.4	85.5 ±1.0	87.8 ±0.9	82.6 ±1.2	13.2 ±2.8
CRT [23]	81.6 ±0.1	70.0 ±0.4	**89.7** ±0.2	90.4 ±0.3	83.9 ±0.5	12.9 ±0.0
BoDA$_r$	78.6 ±0.4	68.5 ±0.3	86.4 ±0.1	85.0 ±1.0	80.0 ±0.9	13.7 ±2.1
BoDA-M$_r$	79.4 ±0.6	71.3 ±0.4	88.4 ±0.3	76.2 ±2.7	88.3 ±1.6	14.4 ±1.4
BoDA$_{r,c}$	82.3 ±0.3	68.5 ±0.6	89.2 ±0.2	**92.5** ±0.9	88.3 ±1.2	21.3 ±0.7
BoDA-M$_{r,c}$	**83.0** ±0.4	**74.6** ±0.7	89.2 ±0.2	91.2 ±0.6	**91.7** ±2.0	**21.7** ±1.4
BoDA *vs.* ERM	**+7.7**	**+7.2**	**+3.6**	**+22.9**	**+25.6**	**+7.3**

Table 6. Results on `DomainNet-MLT`.

	Accuracy (by domain)		Accuracy (by shot)			
Algorithm	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM [46]	58.6 ±0.2	29.4 ±0.3	66.0 ±0.1	56.1 ±0.1	35.9 ±0.5	27.6 ±0.3
IRM [1]	57.1 ±0.1	27.6 ±0.1	64.7 ±0.1	54.3 ±0.3	33.5 ±0.3	25.8 ±0.3
GroupDRO [40]	53.6 ±0.1	25.9 ±0.2	61.8 ±0.1	49.1 ±0.3	30.7 ±0.7	22.0 ±0.1
Mixup [50]	57.6 ±0.1	28.7 ±0.0	64.9 ±0.2	54.5 ±0.1	35.6 ±0.2	27.3 ±0.3
MLDG [27]	58.5 ±0.0	28.7 ±0.1	66.0 ±0.1	55.7 ±0.1	35.3 ±0.2	26.9 ±0.3
CORAL [45]	59.4 ±0.1	30.1 ±0.4	66.4 ±0.1	57.1 ±0.0	37.7 ±0.6	29.9 ±0.2
MMD [29]	56.7 ±0.0	27.2 ±0.2	64.2 ±0.1	54.0 ±0.0	33.9 ±0.2	25.4 ±0.2
DANN [15]	55.8 ±0.1	26.9 ±0.4	63.0 ±0.1	52.7 ±0.1	34.2 ±0.4	26.8 ±0.4
CDANN [31]	56.0 ±0.1	27.7 ±0.1	63.2 ±0.0	52.7 ±0.2	34.3 ±0.5	27.6 ±0.1
MTL [4]	58.6 ±0.1	29.3 ±0.2	65.9 ±0.1	56.0 ±0.4	35.4 ±0.1	28.2 ±0.3
SagNet [35]	58.9 ±0.0	29.4 ±0.2	66.3 ±0.1	56.4 ±0.0	36.2 ±0.3	27.2 ±0.4
Fish [42]	59.6 ±0.1	29.1 ±0.1	67.1 ±0.1	57.2 ±0.1	36.8 ±0.4	27.8 ±0.3
Focal [32]	57.8 ±0.2	27.5 ±0.1	65.2 ±0.2	55.1 ±0.2	35.8 ±0.1	26.3 ±0.1
CBLoss [10]	58.9 ±0.1	30.1 ±0.1	64.3 ±0.0	61.0 ±0.3	42.5 ±0.4	28.1 ±0.2
LDAM [6]	59.2 ±0.0	29.2 ±0.2	66.6 ±0.0	57.0 ±0.0	37.1 ±0.2	27.8 ±0.3
BSoftmax [39]	58.9 ±0.1	29.9 ±0.1	64.3 ±0.1	60.9 ±0.3	42.4 ±0.6	28.2 ±0.1
SSP [52]	59.7 ±0.0	31.6 ±0.2	64.3 ±0.1	62.6 ±0.1	45.0 ±0.3	30.5 ±0.0
CRT [23]	60.4 ±0.2	31.6 ±0.1	66.8 ±0.0	61.6 ±0.1	45.7 ±0.1	29.7 ±0.1
BoDA$_r$	60.1 ±0.2	32.6 ±0.1	65.7 ±0.2	60.6 ±0.1	42.6 ±0.3	30.5 ±0.2
BoDA-M$_r$	60.1 ±0.2	32.2 ±0.2	65.9 ±0.2	60.7 ±0.1	42.9 ±0.3	30.0 ±0.1
BoDA$_{r,c}$	**61.7** ±0.1	**33.4** ±0.1	**67.0** ±0.1	62.7 ±0.1	46.0 ±0.2	**32.2** ±0.3
BoDA-M$_{r,c}$	**61.7** ±0.2	33.3 ±0.1	**67.0** ±0.1	**63.0** ±0.3	**46.6** ±0.4	31.8 ±0.2
BoDA *vs.* ERM	**+3.1**	**+4.0**	**+1.0**	**+6.9**	**+10.7**	**+4.6**

Table 7. Results over all MDLT benchmarks.

Algorithm	`VLCS-MLT`	`PACS-MLT`	`OfficeHome-MLT`	`TerraInc-MLT`	`DomainNet-MLT`	**Avg**
ERM [46]	76.3 ±0.4	97.1 ±0.1	80.7 ±0.0	75.3 ±0.3	58.6 ±0.2	77.6
IRM [1]	76.5 ±0.2	96.7 ±0.2	80.6 ±0.4	73.3 ±0.7	57.1 ±0.1	76.8
GroupDRO [40]	76.7 ±0.4	97.0 ±0.1	80.1 ±0.3	72.0 ±0.4	53.6 ±0.1	75.9
Mixup [50]	75.9 ±0.1	96.7 ±0.2	81.2 ±0.2	71.1 ±0.7	57.6 ±0.1	76.5
MLDG [27]	76.9 ±0.2	96.6 ±0.1	80.4 ±0.2	76.6 ±0.2	58.5 ±0.0	77.8
CORAL [45]	75.9 ±0.5	96.6 ±0.5	81.9 ±0.1	76.4 ±0.5	59.4 ±0.1	78.0
MMD [29]	76.3 ±0.6	96.9 ±0.1	78.4 ±0.4	73.3 ±0.4	56.7 ±0.0	76.3
DANN [15]	77.5 ±0.1	96.5 ±0.0	79.2 ±0.2	68.7 ±0.9	55.8 ±0.1	75.5
CDANN [31]	76.6 ±0.4	96.1 ±0.1	79.0 ±0.2	70.3 ±0.5	56.0 ±0.1	75.6
MTL [4]	76.3 ±0.3	96.7 ±0.2	79.5 ±0.2	75.0 ±0.7	58.6 ±0.1	77.2
SagNet [35]	76.3 ±0.2	**97.2** ±0.1	80.9 ±0.1	75.1 ±1.6	58.9 ±0.0	77.7
Fish [42]	77.5 ±0.3	96.9 ±0.2	81.3 ±0.3	75.3 ±0.5	59.6 ±0.1	78.1
Focal [32]	75.6 ±0.4	96.5 ±0.2	77.9 ±0.0	75.7 ±0.4	57.8 ±0.2	76.7
CBLoss [10]	76.8 ±0.3	96.9 ±0.1	79.8 ±0.2	78.0 ±0.4	58.9 ±0.1	78.1
LDAM [6]	77.5 ±0.1	96.5 ±0.2	80.3 ±0.2	74.7 ±0.9	59.2 ±0.0	77.7
BSoftmax [39]	76.7 ±0.5	96.9 ±0.3	80.4 ±0.2	76.7 ±1.0	58.9 ±0.1	77.9
SSP [52]	76.1 ±0.3	96.9 ±0.2	81.1 ±0.3	78.5 ±0.7	59.7 ±0.0	78.5
CRT [23]	76.3 ±0.2	96.3 ±0.1	81.2 ±0.0	81.6 ±0.1	60.4 ±0.2	79.2
BoDA$_r$	76.9 ±0.5	97.0 ±0.1	81.5 ±0.1	78.6 ±0.4	60.1 ±0.2	78.8
BoDA-M$_r$	77.5 ±0.3	97.1 ±0.1	81.9 ±0.2	79.4 ±0.6	60.1 ±0.2	79.2
BoDA$_{r,c}$	77.3 ±0.2	**97.2** ±0.1	82.3 ±0.1	82.3 ±0.3	**61.7** ±0.1	80.2
BoDA-M$_{r,c}$	**78.2** ±0.4	97.1 ±0.2	**82.4** ±0.2	**83.0** ±0.4	**61.7** ±0.2	**80.5**
BoDA *vs.* ERM	**+1.9**	**+0.1**	**+1.7**	**+7.7**	**+3.1**	**+2.9**

are in Table 2, 3, 4, 5 and 6, respectively. We highlight rows in gray for `BoDA` and its variants, and bolden the best result in each column. First, as all tables indicate, `BoDA` consistently achieves the best average accuracy across all datasets. It also achieves the best worst-case accuracy most of the time. Moreover, on certain datasets (e.g., `OfficeHome-MLT`), MDL methods perform better (e.g., CORAL), while on others (e.g., `TerraInc-MLT`), imbalanced methods achieve higher gains (e.g., CRT); Nevertheless, regardless of dataset, `BoDA` outperforms all methods, highlighting its effectiveness for the MDLT task. Finally, compared to ERM, `BoDA` slightly improves the average and many-shot performance, while substantially boosting the performance for the medium-shot, few-shot, and zero-shot pairs. Table 7 summarizes the averaged accuracy across all datasets, where `BoDA` brings large overall improvements of $\sim 3\%$.

A Closer Look at Accuracy Gains. We further explore how `BoDA` performs across *all* domain-class pairs. Fig. 7 shows the absolute accuracy gains of `BoDA` over ERM on `OfficeHome-MLT`, where `BoDA` consistently improves the performance over all domains. The improvements are especially large for domain "Art", where most of the classes lie in the *few-shot* region. For certain classes, `BoDA` can improve up to 50% accuracy, indicating its effectiveness on tackling MDLT.

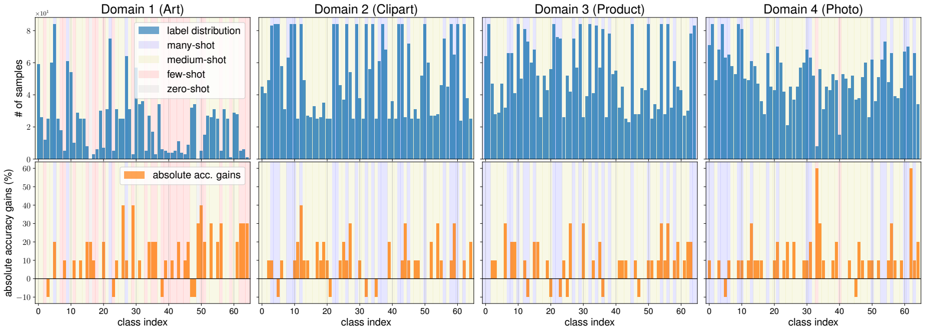


Fig. 7. The absolute accuracy gains of `BoDA` *vs.* ERM over all domain-class pairs on `OfficeHome-MLT`. `BoDA` establishes large improvements w.r.t. all regions, especially for the few-shot and zero-shot ones. Results for other datasets are in Appendix H.2.

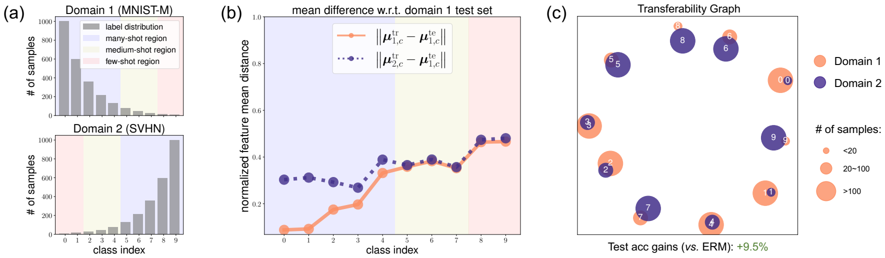


Fig. 8. `BoDA` analysis. **(a)** Label distribution setup. **(b)** Distance of feature mean between train and test data. `BoDA` enables better learned tail (d, c) with smaller feature discrepancy. **(c)** `BoDA` learns features that are more aligned across domains even in the presence of divergent labels, and significantly improves upon ERM by 9.5%.

Ablation Studies on `BoDA` Components (Appendix H.1). We study the effects of (1) adding balanced distance (i.e., `BoDA` *vs.* vanilla `DA`), and (2) different choices of distance calibration coefficient $\lambda_{d,c}^{d',c'}$ in `BoDA`. We observe that `BoDA` improves over `DA` by a large margin (2.3% on average over all MDLT datasets), highlighting the importance of using *balanced* distance. Interestingly, as for $\lambda_{d,c}^{d',c'}$, we find that `BoDA` is pretty robust to different choices within a given range, and obtain similar gains (1.9% to 2.9% over ERM).

6.2 Understanding the Behavior of `BoDA` on MDLT

To better understand how the design of `BoDA` contributes to its superior performance, we revisit the `Digits-MLT` dataset and run `BoDA` as opposed to ERM.

Better Learned Representations for Minority Data. Similar to Fig. 5, we plot in Fig. 8b the feature mean distance between training and test data for `BoDA` on `Digits-MLT`. The plot shows that, in `BoDA`, the distance between training and test data for minority classes (class "8" and "9") becomes smaller.

Improved Transferability against Severe Imbalance. Fig. 8c plots the transferability graph induced by `BoDA` for the label distributions in Fig. 8a. It shows that even in the presence of severe and divergent label imbalance, `BoDA`

Table 9. BoDA strengthens performance on Domain Generalization (DG) benchmarks. The full tables including detailed results for each dataset are in Appendix G.

Algorithm	VLCS	PACS	OfficeHome	TerraInc	DomainNet	**Avg**
ERM	77.5 ±0.4	85.5 ±0.2	66.5 ±0.3	46.1 ±1.8	40.9 ±0.1	63.3
Current SOTA [45]	**78.8** ±0.6	86.2 ±0.3	68.7 ±0.3	47.6 ±1.0	41.5 ±0.1	64.5
BoDA$_{r,c}$	78.5 ±0.3	**86.9** ±0.4	**69.3** ±0.1	**50.2** ±0.4	**42.7** ±0.1	**65.5**
BoDA$_{r,c}$ + Current SOTA [45]	79.1 ±0.1	87.9 ±0.5	69.9 ±0.2	50.7 ±0.6	43.5 ±0.3	66.2
BoDA *vs.* ERM	**+1.6**	**+2.4**	**+3.4**	**+4.6**	**+2.6**	**+2.9**

learns transferable features. Further, BoDA learns a *balanced* feature space that separates different classes away. The better learned features translates to better accuracy (9.5% gains vs. ERM in Fig. 3c). More results are in Appendix H.3.

Tightness of the Bound. We study whether the BoDA bound derived in Theorem 1 is tight. We train a ResNet-18 on Digits-MLT for 5,000 steps to ensure convergence. We compute the loss over all samples, and combine the results over 3 random seeds. Table 8 confirms the bound is empirically tight.

Table 8. BoDA bound.

	$\mathcal{L}_{\texttt{BoDA}}$
Empirical	2.92947 ±7.3e-3
Theoretical	2.92513 ±7.8e-3

7 Beyond MDLT: (Imbalanced) Domain Generalization

Domain Generalization (DG) refers to learning from multiple domains and generalizing to unseen domains. Since naturally the learning domains differ in their label distributions and may even have class imbalance within each domain, we study whether BoDA can improve performance for DG. Note that all datasets we adapted for MDLT are standard benchmarks for DG, which confirms that data imbalance is an intrinsic problem in DG, but has been overlooked by past works.

To test BoDA, we follow the DG evaluation protocol in [19], and compare to the current SOTA [45]. Table 9 reveals the following findings: First, BoDA alone can improve upon the current SOTA on four out of the five datasets, and achieves notable average performance gains. Moreover, combined with the current SOTA, BoDA further boosts the result by a notable margin across all datasets, suggesting that label imbalance is orthogonal to existing DG-specific algorithms. Finally, similar to MDLT, the gains depend on how severe the imbalance is within a dataset – e.g., TerraInc exhibits the most severe label imbalance across domains, on which BoDA achieves the highest gains. The intriguing results shed light on the importance of integrating label imbalance for practical DG algorithm design.

8 Conclusion

We formalize MDLT as learning from multi-domain imbalanced data, and generalizing to all domain-class pairs. We introduce the domain-class transferability graph, and propose BoDA, a theoretically grounded loss that tackles MDLT. Extensive results on real-world MDLT benchmarks verify its superiority. Furthermore, BoDA establishes a new SOTA on DG benchmarks. Our work opens up new avenues for realistic multi-domain learning in the presense of data imbalance.

Acknowledgments. This work was supported by the GIST-MIT Research Collaboration grant funded by GIST.

References

1. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019)
2. Beery, S., Van Horn, G., Perona, P.: Recognition in terra incognita. In: ECCV (2018)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1), 151–175 (2010)
4. Blanchard, G., Deshmukh, A.A., Dogan, U., Lee, G., Scott, C.: Domain generalization by marginal transfer learning. Journal of Machine Learning Research **22**(2), 1–55 (2021)
5. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018)
6. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: NeurIPS (2019)
7. Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain generalization by solving jigsaw puzzles. In: CVPR (2019)
8. Carroll, J.D., Arabie, P.: Multidimensional scaling. Measurement, judgment and decision making pp. 179–250 (1998)
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
10. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR (2019)
11. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L.: The mahalanobis distance. Chemometrics and intelligent laboratory systems **50**(1), 1–18 (2000)
12. Dong, Q., Gong, S., Zhu, X.: Imbalanced deep learning by minority class incremental rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(6), 1367–1381 (2019)
13. Dredze, M., Kulesza, A., Crammer, K.: Multi-domain learning by confidence-weighted parameter combination. Machine Learning **79**(1), 123–149 (2010)
14. Fang, C., Xu, Y., Rockmore, D.N.: Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In: ICCV (2013)
15. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of machine learning research **17**(1), 2096–2030 (2016)
16. Globerson, A., Chechik, G., Pereira, F., Tishby, N.: Euclidean embedding of co-occurrence data. In: NeurIPS (2004)
17. Goldberger, J., Hinton, G.E., Roweis, S., Salakhutdinov, R.R.: Neighbourhood components analysis. In: NeurIPS (2004)
18. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1), 723–773 (2012)
19. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. In: ICLR (2021)
20. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE international joint conference on neural networks. pp. 1322–1328 (2008)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

22. Huang, C., Li, Y., Chen, C.L., Tang, X.: Deep imbalanced learning for face recognition and attribute prediction. IEEE transactions on pattern analysis and machine intelligence (2019)
23. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. ICLR (2020)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
25. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Priol, R.L., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). arXiv preprint arXiv:2003.00688 (2020)
26. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
27. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Learning to generalize: Meta-learning for domain generalization. In: AAAI (2018)
28. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: ICCV (2017)
29. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: CVPR (2018)
30. Li, T., Cao, P., Yuan, Y., Fan, L., Yang, Y., Feris, R., Indyk, P., Katabi, D.: Targeted supervised contrastive learning for long-tailed recognition. arXiv preprint arXiv:2111.13998 (2021)
31. Li, Y., Gong, M., Tian, X., Liu, T., Tao, D.: Domain generalization via conditional invariant representations. In: AAAI (2018)
32. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
33. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR (2019)
34. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML (2013)
35. Nam, H., Lee, H., Park, J., Yoon, W., Yoo, D.: Reducing domain gap by reducing style bias. In: CVPR (2021)
36. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
37. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
38. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV (2019)
39. Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al.: Balanced meta-softmax for long-tailed visual recognition. In: NeurIPS (2020)
40. Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In: ICLR (2020)
41. Schoenauer-Sebag, A., Heinrich, L., Schoenauer, M., Sebag, M., Wu, L.F., Altschuler, S.J.: Multi-domain adversarial learning. In: ICLR (2019)
42. Shi, Y., Seely, J., Torr, P.H., Siddharth, N., Hannun, A., Usunier, N., Synnaeve, G.: Gradient matching for domain generalization. arXiv preprint arXiv:2104.09937 (2021)
43. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting. arXiv preprint arXiv:1902.07379 (2019)

44. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: NeurIPS (2016)
45. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV (2016)
46. Vapnik, V.N.: An overview of statistical learning theory. IEEE transactions on neural networks **10**(5), 988–999 (1999)
47. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR (2017)
48. Wang, X., Lian, L., Miao, Z., Liu, Z., Yu, S.: Long-tailed recognition by routing diverse distribution-aware experts. In: ICLR (2021)
49. Xiao, T., Li, H., Ouyang, W., Wang, X.: Learning deep feature representations with domain guided dropout for person re-identification. In: CVPR (2016)
50. Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., Zhang, W.: Adversarial domain adaptation with domain mixup. In: AAAI (2020)
51. Yang, Y., Hospedales, T.M.: A unified perspective on multi-domain and multi-task learning. In: ICLR (2015)
52. Yang, Y., Xu, Z.: Rethinking the value of labels for improving class-imbalanced learning. In: NeurIPS (2020)
53. Yang, Y., Zha, K., Chen, Y.C., Wang, H., Katabi, D.: Delving into deep imbalanced regression. In: ICML (2021)
54. Zhang, M., Marklund, H., Gupta, A., Levine, S., Finn, C.: Adaptive risk minimization: A meta-learning approach for tackling group shift. arXiv preprint arXiv:2007.02931 (2020)
55. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: ICCV (2017)
56. Zhang, Y., Hooi, B., Hong, L., Feng, J.: Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. arXiv preprint arXiv:2107.09249 (2021)
57. Zhang, Y., Kang, B., Hooi, B., Yan, S., Feng, J.: Deep long-tailed learning: A survey. arXiv preprint arXiv:2110.04596 (2021)
58. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. CVPR (2020)
59. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization in vision: A survey. arXiv preprint arXiv:2103.02503 (2021)
60. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: ICLR (2021)