Moment Matching for Multi-Source Domain Adaptation

Xingchao Peng\(^1\), Qinxun Bai\(^1\), Xide Xia\(^1\), Zijun Huang\(^2\), Kate Saenko\(^1\), Bo Wang\(^3\)

xpeng,qinxun,xidexia,saenko@bu.edu, zijun.huang@columbia.edu, bowang@vectorinstitute.ai

\(^1\)Department of Computer Science, Boston University
\(^2\)Columbia University & Mado AI Research
\(^3\)Vector Institute for Artificial Intelligence

Abstract

Conventional unsupervised domain adaptation (UDA) assumes that training data are sampled from a single domain. This neglects the more practical scenario where training data are collected from multiple sources, requiring multi-source domain adaptation. We make three major contributions towards addressing this problem. First, we collect and annotate by far the largest UDA dataset with six domains and about 0.6 million images distributed among 345 categories, addressing the gap in data availability for multi-source UDA research. Second, we propose a new deep learning approach, Moment Matching for Multi-Source Domain Adaptation (M\(^3\)SDA), which aims to transfer knowledge learned from multiple labeled source domains to an unlabeled target domain by dynamically aligning moments of their feature distributions. Third, we provide new theoretical insight specifically for moment matching approaches in both single and multi-source domain adaptation. Extensive experiments are conducted to demonstrate the power of our new dataset in benchmarking state-of-the-art multi-source domain adaptation methods, as well as the advantage of our proposed model. Data and source code are available at http://ai.bu.edu/M3SDA/.

1. Introduction

Generalizing models learned on one visual domain to novel domains has been a major obstacle in the quest for universal object recognition. The performance of the learned models degrades significantly when testing on novel domains due to the presence of domain shift [36].

Recently, transfer learning and domain adaptation methods have been proposed to mitigate the domain gap. For example, several UDA methods [27, 41, 25] incorporate Maximum Mean Discrepancy loss into a neural network to diminish the domain discrepancy; other models introduce different learning schema to align the source and target domains, including aligning second order correlation [39, 32], moment matching [47], adversarial domain confusion [40, 8, 38] and GAN-based alignment [50, 15, 23]. However, most of current UDA methods assume that source samples are collected from a single domain. This assumption neglects the more practical scenarios where labeled images are typically collected from multiple domains. For example, the training images can be taken under different weather or lighting conditions, share different visual cues, and even have different modalities (as shown in Figure 1).

In this paper, we consider multi-source domain adaptation (MSDA), a more difficult but practical problem of knowledge transfer from multiple distinct domains to one unlabeled target domain. The main challenges in the research of MSDA are that: (1) the source data has multiple domains, which hampers the effectiveness of mainstream single UDA method; (2) source domains also possess domain shift with each other; (3) the lack of large-scale multi-domain dataset hinders the development of MSDA models.

In the context of MSDA, some theoretical analysis [1, 28, 4, 49, 14] has been proposed for multi-source domain adaptation (MSDA). Ben-David et al [1] pioneer this direction by introducing an \(\mathcal{H}\Delta\mathcal{H}\)-divergence between the weighted combination of source domains and target domain. More applied works [6, 45] use an adversarial dis-
2. Related Work

We first summarize the state-of-the-art domain adaptation benchmarks and explain how our LSDAC differs from previous ones. Then, we describe related domain adaptation methods.

**Domain Adaptation Datasets** Several notable datasets that can be utilized to evaluate domain adaptation approaches are summarized in Table 1. The Office dataset [37] is a popular benchmark for office environment objects. It contains 31 categories captured in three domains: office environment images taken with a high quality camera (DSLR), office environment images taken with a low quality camera (Webcam), and images from an online merchandising website (Amazon). The office dataset and its extension, Office-Caltech10 [11], have been used in numerous domain adaptation papers [25, 40, 27, 39, 45], and the adaptation performance has reached ~90% accuracy. More recent benchmarks [43, 17, 34] are proposed to evaluate the effectiveness of domain adaptation models. However, these datasets are small-scale and limited by their specific environments, such as office, home, and museum. Our dataset contains about 600k images, distributed in 345 categories and 6 distinct domains. We capture various object divisions, ranging from furniture, cloth, electronic to mammal, building, etc.

**Single-sourceUDA** Over the past decades, various single-source UDA methods have been proposed. These methods can be taxonomically divided into three categories. The first category is the discrepancy-based DA approach, which utilizes different metric learning schemas to diminish the domain shift between source and target domains. Inspired by the kernel two-sample test [12], Maximum Mean Discrepancy (MMD) is applied to reduce distribution shift in various methods [27, 41, 9, 44]. Other commonly used methods include correlation alignment [39, 32], Kullback-Leibler (KL) divergence [51], and $\mathcal{H}$ divergence [1]. The second category is the adversarial-based approach [24, 40]. A domain discriminator is leveraged to encourage the domain confusion by an adversarial objective. Among these approaches, generative adversarial networks are widely used to learn domain-invariant features as well to generate fake source or target data. Other frameworks utilize only adversarial loss to bridge two domains. The third category is reconstruction-based, which assumes the data reconstruction helps the DA models to learn domain-invariant features. The reconstruction is obtained via an encoder-decoder [3, 10] or a GAN discriminator, such as dual-

| Dataset          | Year | Images   | Classes | Domains |
|------------------|------|----------|---------|---------|
| Digit-Five       | -    | ~100,000 | 10      | 5       |
| Office [37]      | 2010 | 4,110    | 31      | 3       |
| Office-Caltech10 | 2012 | 2,533    | 10      | 4       |
| CAD-Pascal [33]  | 2015 | 12,000   | 20      | 6       |
| Office-Home [43] | 2017 | 15,500   | 65      | 4       |
| PACS [21]        | 2017 | 9,991    | 7       | 4       |
| Open MIC [17]    | 2018 | 16,156   | -       | -       |
| Syn2Real [35]    | 2018 | 280,157  | 12      | 3       |
| LSDAC (Ours)     | -    | 569,010  | 345     | 6       |

Table 1. A collection of most notable datasets to evaluate domain adaptation methods. Specifically, “Digit-Five” dataset indicates five most popular digit datasets (MNIST [19], MNIST-M [8] USPS, Synthetic Digits [8], SVHN, and USPS) which are widely used to evaluate domain adaptation models. Our dataset is challenging as it contains more images, categories, and domains than other datasets. (see Table 9, Table 10, and Table 11 in Appendix for detailed categories.)
GAN [46], cycle-GAN [50], disco-GAN [16], and Cy-CADA [15]. Though these methods make progress on UDA, few of them consider the practical scenario where training data are collected from multiple sources. Our paper proposes a model to tackle multi-source domain adaptation, which is a more general and challenging scenario.

**Multi-Source Domain Adaptation** Compared with single source UDA, multi-source domain adaptation assumes that training data from multiple sources are available. Originated from the early theoretical analysis [1, 28, 4], MSDA has many practical applications [45, 6]. Ben-David et al [1] introduce an $H \Delta H$-divergence between the weighted combination of source domains and target domain. Crammer et al [4] establish a general bound on the expected loss of the model by minimizing the empirical loss on the nearest k sources. Mansour et al [28] claim that the target hypothesis can be represented by a weighted combination of source hypotheses. In the more applied works, Deep Cocktail Network (DCTN) [45] proposes a k-way domain discriminator and category classifier for digit classification and real-world object recognition. Hoffman et al [14] propose normalized solutions with theoretical guarantees for cross-entropy loss, aiming to provide a solution for the MSDA problem with very practical benefits. Duan et al [6] propose Domain Selection Machine for event recognition in consumer videos by leveraging a large number of loosely labeled web images from different sources. Different from these methods, our model directly matches all the distributions by matching the moments. Moreover, we provide a concrete proof of why matching the moments of multiple distributions works for multi-source domain adaptation.

**Moment Matching** The moments of distributions have been studied by the machine learning community for a long time. In order to diminish the domain discrepancy between two domains, different moment matching schemes have been proposed. For example, MMD matches the first moments of two distributions. Sun et al [39] propose an approach that matches the second moments. Zhang et al [48] propose to align infinite-dimensional covariance matrices in RKHS. Zellinger et al [47] introduce a moment matching regularizer to match high moments. As the generative adversarial network (GAN) becomes popular, many GAN-based moment matching approaches have been proposed. MGAN [29] utilizes a GAN to match the mean and covariance of feature distributions. GMMN [22] and MMD GAN [20] are proposed for aligning distribution moments with generative neural networks. Compared to these methods, our work focuses on matching distribution moments for multiple domains and more importantly, we demonstrate that this is crucial for multi-source domain adaptation.

## 3. The LSDAC dataset

It is well-known that deep models require massive amounts of training data. Unfortunately, existing datasets for visual domain adaptation are either small-scale or limited in the number of categories. We collect by far the largest domain adaptation dataset to date, LSDAC (Large Scale Domain Adaptation Challenge). The LSDAC contains six domains, with each domain containing 345 categories of common objects, as listed in Table 9, Table 10, and Table 11 (see Appendix). The domains include Clipart (clp, see Appendix, Figure 8); collection of clipart images; Infograph (inf, see Figure 9); infographic images with specific object; Painting (pmt, see Figure 10): artistic depictions of objects in the form of paintings; Quickdraw (qdr, see Figure 11): drawings of the worldwide players of game “Quick Draw”\(^1\); Real (rel, see Figure 12): photos.

\(^1\)https://quickdraw.withgoogle.com/data
Figure 3. The framework of Moment Matching for Multi-source Domain Adaptation (M^3SDA). Our model consists of three components: i) feature extractor, ii) moment matching component, and iii) classifiers. Our model takes multi-source annotated training data as input and transfers the learned knowledge to classify the unlabeled target samples. Without loss of generality, we show the i-th domain and j-th domain as an example. The feature extractor maps the source domains into a common feature space. The moment matching component attempts to match the i-th and j-th domains with the target domain, as well as matching the i-th domain with the j-th domain. The final predictions of target samples are based on the weighted outputs of the i-th and j-th classifiers. (Best viewed in color!)

tos and real world images; and Sketch (skt, see Figure 13): sketches of specific objects.

The images from clipart, infograph, painting, real, and sketch domains are collected by searching a category name combined with a domain name (e.g. “aeroplane painting”) in different image search engines. One of the main challenges is that the downloaded data contain a large portion of outliers. To clean the dataset, we hire 20 annotators to manually filter out the outliers. This process took around 2,500 hours (more than 2 weeks) in total. To control the annotation quality, we assign two annotators to each image, and only take the images agreed by both annotators. After the filtering process, we keep 423.5k images from the 1.2 million images crawled from the web. The dataset has an average of around 150 images per category for clipart and infograph domain, around 220 per category for painting and sketch domain, and around 510 for real domain. A statistical overview of the dataset is shown in Figure 2.

The quickdraw domain is downloaded directly from https://quickdraw.withgoogle.com/. The raw data are presented as a series of discrete points with temporal information. We use the B-spline [5] algorithm to connect all the points in each strike to get a complete drawing. We choose 500 images for each category to form the quickdraw-drawn, which contains 172.5k images in total.

4. Moment Matching for Multi-Source DA

Given $D_S = \{D_1, D_2, ..., D_N\}$ the collection of labeled source domains and $D_T$ the unlabeled target domain, where all domains are defined by bounded rational measures on input space $\mathcal{X}$, the multi-source domain adaptation problem aims to find a hypothesis in the given hypothesis space $\mathcal{H}$, which minimizes the testing target error on $D_T$.

Definition 1. Assume $X_1, X_2, ..., X_N, X_T$ are collections of i.i.d. samples from $D_1, D_2, ..., D_N, D_T$ respectively, then the Moment Distance between $D_S$ and $D_T$ is defined as

$$MD^2(D_S, D_T) = \frac{2}{N^2} \sum_{k=1}^{N} \left( \frac{1}{N} \sum_{i=1}^{N} \|E(X_i^k) - E(X_T^k)\|_2^2 \right)$$

$$+ \left( \frac{N}{2} \right)^{-1} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \|E(X_i^k) - E(X_j^k)\|_2^2. \tag{1}$$

M^3SDA We propose a moment-matching model for MSDA based on deep neural networks. As shown in Figure 3, our model comprises of a feature extractor $G$, a moment-matching component, and a set of $N$ classifiers $C = \{C_1, C_2, ..., C_N\}$. The feature extractor $G$ maps $D_S, D_T$ to a common latent feature space. The moment matching component minimizes the moment-related distance defined in Equation 1. The $N$ classifiers are trained on the annotated source domains with cross-entropy loss. The overall objective function is:

$$\min_{G, C} \sum_{i=1}^{N} L_{D_i} + \lambda \min_{G} MD^2(D_S, D_T), \tag{2}$$

where $L_{D_i}$ is the softmax cross entropy loss for the classifier $C_i$ on domain $D_i$, and $\lambda$ is the trade-off parameter.

M^3SDA-β In order to align $p(y|x)$ and $p(x)$ at the same time, we follow the training paradigm proposed by [38]. In particular, we leverage two classifiers per domain to form $N$ pairs of classifiers $C' = \{C_{i,1}, C_{i,2}\}$. The feature extractor $G$ is trained to minimize the moment distance $MD^2(D_S, D_T)$, which is defined as:

$$MD^2(D_S, D_T) = \frac{2}{N^2} \sum_{k=1}^{N} \left( \frac{1}{N} \sum_{i=1}^{N} \|E(X_i^k) - E(X_T^k)\|_2^2 \right)$$

$$+ \left( \frac{N}{2} \right)^{-1} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \|E(X_i^k) - E(X_j^k)\|_2^2. \tag{1}$$

M^3SDA-β Our model consists of three components: i) feature extractor, ii) moment matching component, and iii) classifiers. Our model takes multi-source annotated training data as input and transfers the learned knowledge to classify the unlabeled target samples. Without loss of generality, we show the i-th domain and j-th domain as an example. The feature extractor maps the source domains into a common feature space. The moment matching component attempts to match the i-th and j-th domains with the target domain, as well as matching the i-th domain with the j-th domain. The final predictions of target samples are based on the weighted outputs of the i-th and j-th classifiers. (Best viewed in color!)
\{(C_1, C'_1), (C_2, C'_2), ..., (C_N, C'_N)\}. The training procedure includes three steps. i) We train \(G\) and \(C'\) to classify the multi-source samples correctly. The objective is similar to Equation 2. ii). We then train the classifier pairs for a fixed \(G\). The goal is to make the discrepancy of each pair of classifiers as large as possible on the target domain. For example, the outputs of \(C_1\) and \(C'_1\) should possess a large discrepancy. Following [38], we define the discrepancy of two classifiers as the L1-distance between the outputs of the two classifiers. The objective is:

\[
\min_{\alpha} \sum_{i=1}^{N} L_{D_i} - \sum_{i} |P_{C_i}(D_T) - P_{C'_i}(D_T)|, \tag{3}
\]

where \(P_{C_i}(D_T), P_{C'_i}(D_T)\) denote the outputs of \(C_i, C'_i\) respectively on the target domain. iii). Finally, we fix \(C'\) and train \(G\) to minimize the discrepancy of each classifier pair on the target domain. The objective function is as follows:

\[
\min_{G} \sum_{i} |P_{C_i}(D_T) - P_{C'_i}(D_T)|, \tag{4}
\]

These three training steps are performed periodically until the whole network converges.

**Ensemble Schema** In the testing phase, testing data from the target domain are forwarded through the feature generator and the \(N\) classifiers. We propose two schemas to combine the outputs of the classifiers:

- average the outputs of the classifiers, marked as M3SDA*.
- Derive a weight vector \(W = (w_1, \ldots, w_{N-1})\) \((\sum_{j=1}^{N-1} w_j = 1\), assuming \(N\)-th domain is the target\). The final prediction is the weighted average of the outputs.

To this end, how to derive the weight vector becomes a critical problem. The main philosophy of the weight vector is to make it represent the intrinsic closeness between the target domain and source domains. In our setting, the weighted vector is derived by the source-only accuracy between the \(i\)-th domain and the \(N\)-th domain, i.e., \(w_i = \text{acc}_i/\sum_{j=1}^{N-1} \text{acc}_j\).

### 4.1. Theoretical Insight

Following [1], we introduce a rigorous model of multi-source domain adaptation for binary classification. A domain \(D = (\mu, f)\) is defined by a probability measure (distribution) \(\mu\) on the input space \(\mathcal{X}\) and a labeling function \(f: \mathcal{X} \rightarrow \{0, 1\}\). A hypothesis is a function \(h: \mathcal{X} \rightarrow \{0, 1\}\). The probability that \(h\) disagrees with the domain labeling function \(f\) under the domain distribution \(\mu\) is defined as:

\[
\epsilon_D(h) = \epsilon_D(h, f) = \mathbb{E}_\mu[||h(x) - f(x)||]. \tag{5}
\]

For a source domain \(D_S\) and a target domain \(D_T\), we refer to the source error and the target error of a hypothesis \(h\) as \(\epsilon_S(h) = \epsilon_{D_S}(h)\) and \(\epsilon_T(h) = \epsilon_{D_T}(h)\) respectively. When the expectation in Equation 5 is computed with respect to an empirical distribution, we denote the corresponding empirical error by \(\hat{\epsilon}_D(h)\), such as \(\hat{\epsilon}_S(h)\) and \(\hat{\epsilon}_T(h)\). In particular, we examine algorithms that minimize convex combinations of source errors, i.e., given a weight vector \(\alpha = (\alpha_1, \ldots, \alpha_N)\) with \(\sum_{j=1}^{N} \alpha_j = 1\), we define the \(\alpha\)-weighted source error of a hypothesis \(h\) as \(\epsilon_{\alpha}(h) = \sum_{j=1}^{N} \alpha_j \epsilon_j(h)\), where \(\epsilon_j(h)\) is the shorthand of \(\epsilon_{D_j}(h)\). The empirical \(\alpha\)-weighted source error can be defined analogously and denoted by \(\hat{\epsilon}_{\alpha}(h)\).

Previous theoretical bounds [1, 14, 49] on the target error are based on the \(\mathcal{H}\Delta\mathcal{H}\)-divergence between the source and target domains. While providing theoretical insights for general multi-source domain adaptation, these \(\mathcal{H}\Delta\mathcal{H}\)-divergence based bounds do not directly motivate moment-based approaches. In order to provide a specific insight for moment-based approaches, we introduce the \(k\)-th order cross-moment divergence between domains, denoted by \(d_{CM^k}(\cdot, \cdot)\), and extend the analysis in [1] to derive the following moment-based bound for multi-source domain adaptation. See Appendix for the definition of the cross-moment divergence and the proof of the theorem.

**Theorem 1.** Let \(\mathcal{H}\) be a hypothesis space of VC dimension \(d\). Let \(m\) be the size of labeled samples from all sources \(\{D_1, D_2, ..., D_N\}\), \(S_j\) be the labeled sample set of size \(\beta_j m\) \((\sum_{j} \beta_j = 1)\) drawn from \(\mu_j\) and labeled by the groundtruth labeling function \(f_j\). If \(h \in \mathcal{H}\) is the empirical minimizer of \(\hat{\epsilon}_\alpha(h)\) for a fixed weight vector \(\alpha\) and \(\beta_1^* = \min_{h \in \mathcal{H}} \epsilon_T(h)\) is the target error minimizer, then for any \(\delta \in (0, 1)\) and any \(\epsilon > 0\), there exist \(N\) integers \(\{n_j\}_{j=1}^{N}\) and \(N\) constants \(\{a_{n_j}\}_{j=1}^{N}\), such that with probability at least \(1 - \delta\),

\[
\epsilon_T(h) \leq \epsilon_T(h_*) + \eta_{\alpha, \beta, m, \delta} + \epsilon \nonumber + \sum_{j=1}^{N} \alpha_j \left(2\lambda_j + a_{n_j} \sum_{k=1}^{n_j} d_{CM^k}(D_j, D_T)\right), \tag{6}
\]

where \(\eta_{\alpha, \beta, m, \delta} = 4 \sqrt{\left(\sum_{j=1}^{N} \alpha_j^2 \right)^{\lambda/2} \left(3d(\log(\frac{dm}{\lambda}) + 1) + 2 \log(\frac{m}{\delta})\right)}\) and \(\lambda_j = \min_{h \in \mathcal{H}} \{\epsilon_T(h) + \epsilon_j(h)\}\).

Theorem 1 shows that the upper bound on the target error of the learned hypothesis depends on the pairwise moment divergence \(d_{CM^k}(D_S, D_T)\) between the target domain and each source domain.\(^2\) This provides a direct motivation for moment matching approaches beyond ours. In particular, it motivates our multi-source domain adaptation approach to align the moments between each target-source

\(^2\)Note that single source is just a special case when \(N = 1\).
pair. Moreover, it is obvious that the last term of the bound, $\sum_k d_{CM}^k(D_1, D_T)$, is lower bounded by the pairwise divergences between source domains. To see this, consider the toy example consisting of two sources $D_1, D_2$, and a target $D_T$, since $d_{CM}^k(\cdot, \cdot)$ is a metric, triangle inequality implies the following lower bound:

$$d_{CM}^k(D_1, D_T) + d_{CM}^k(D_2, D_T) \geq d_{CM}^k(D_1, D_2).$$

This motivates our algorithm to also align the moments between each pair of source domains. Further discussions of Theorem 1 and its relationship with our algorithm are provided in the Appendix.

5. Experiments

We perform an extensive evaluation on the following tasks: digit classification (MNIST, USPS, SVHN, Synthetic Digits), Office-Caltech10, and LSDAC dataset. In total, we conduct 714 experiments. The experiments are run on a GPU-cluster with 24 GPUs and the total running time is more than 21,440 GPU-hours. Due to space limitations, we only report major results; more implementation details are provided in the supplementary material. All of our experiments are implemented in the PyTorch\(^3\) platform.

5.1. Experiments on Digit Recognition

Five digit datasets are sampled from five different sources, namely MNIST [19], Synthetic Digits [8], MNIST-M [8], SVHN, and USPS. Following DCTN [45], we sample 25000 images from training subset and 9000 from testing subset in MNIST, MNIST-M, SVHN, and Synthetic Digits. USPS dataset contains only 9298 images in total, so we take the entire dataset as a domain. In all of our experiments, we take turns to set one domain as the target domain and the rest as the source domains.

We take four state-of-the-art discrepancy-based approaches: Deep Adaptation Network [25] (DAN), Joint Adaptation Network (JAN), Manifold Embedded Distribution Alignment (MEDA), and Correlation Alignment [39] (CORAL), and four adversarial-based approaches: Domain Adversarial Neural Network [8] (DANN), Adversarial Discriminative Domain Adaptation [40] (ADDA), Maximum Classifier Discrepancy (MCD) and Deep Cocktail Network [45] (DCTN) as our baselines. In the source combine setting, all the source domains are combined to a single domain, and the baseline experiments are conducted in a traditional single domain adaptation manner.

The results are shown in Table 2. Our model $M^{S\text{DA}}$ achieves an 86.13% average accuracy, and $M^{S\text{DA}-\text{\beta}}$ boosts the performance to 87.65%, outperforming other baselines by a large margin. For a fair comparison, all the experiments are based on the same network architecture. For each experiment, we run the same setting for five times and report the mean and standard deviation. (See Appendix for detailed experiment settings and analyses.)

5.2. Experiments on Office-Caltech10

The Office-Caltech10 [11] dataset is extended from the standard Office31 [37] dataset. It consists of the same 10 object categories from 4 different domains: Amazon, Caltech, DSLR, and Webcam.

The experimental results on Office-Caltech10 dataset are shown in Table 4. Our model $M^{S\text{DA}}$ gets a 96.1% average accuracy on this dataset, and $M^{S\text{DA}-\text{\beta}}$ further boosts the performance to 96.4%. All the experiments are based on ResNet-101 pre-trained on ImageNet. As far as we know, our models achieve the best performance among all the results ever reported on this dataset. We have also tried AlexNet, but it did not work as well as ResNet-101.

| Standards | Models | $mm, up, sv, sy$ | $mm, up, sv, sy$ | $mm, mt, sy$ | $mm, mt, up, sy$ | $mm, mt, up, sy$ | Avg |
|-----------|--------|----------------|----------------|-------------|----------------|----------------|-----|
| Source Combine | Source Only | 63.70±0.83 | 92.30±0.91 | 90.71±0.54 | 71.51±0.75 | 83.44±0.79 | 80.33±0.76 |
| | DAN [25] | 67.87±0.75 | 75.50±0.62 | 93.49±0.85 | 67.80±0.84 | 86.93±0.93 | 82.72±0.79 |
| | DANN [8] | 70.81±0.94 | 97.90±0.83 | 93.47±0.79 | 68.50±0.85 | 87.37±0.68 | 83.61±0.82 |
| Multi-Source | Source Only | 63.37±0.74 | 90.50±0.83 | 88.71±0.89 | 63.54±0.93 | 82.44±0.65 | 77.71±0.81 |
| | DAN [25] | 63.78±0.71 | 96.31±0.54 | 94.24±0.71 | 62.45±0.72 | 85.43±0.77 | 80.44±0.72 |
| | CORAL [39] | 62.53±0.69 | 97.21±0.83 | 93.45±0.82 | 64.40±0.72 | 82.77±0.69 | 80.07±0.75 |
| | DANN [8] | 71.50±0.56 | 97.60±0.75 | 92.33±0.85 | 63.48±0.79 | 85.34±0.84 | 82.01±0.76 |
| | JAN [27] | 65.88±0.68 | 97.21±0.73 | 95.42±0.77 | 75.27±0.71 | 86.55±0.64 | 84.07±0.71 |
| | ADDA [40] | 71.57±0.52 | 97.89±0.84 | 92.83±0.74 | 75.48±0.48 | 86.45±0.62 | 84.84±0.64 |
| | DCTN [45] | 70.53±1.24 | 96.23±0.82 | 92.81±0.27 | 77.61±0.41 | 86.77±0.78 | 84.79±0.72 |
| | MEDA [44] | 71.31±0.75 | 96.47±0.78 | 97.01±0.82 | 78.45±0.77 | 84.62±0.79 | 85.60±0.78 |
| | MCD [38] | 72.50±0.67 | 96.21±0.81 | 95.33±0.74 | 78.49±0.78 | 87.47±0.65 | 86.10±0.73 |
| | $M^{S\text{DA}}$ (ours) | 69.76±0.86 | 98.58±0.47 | 95.23±0.79 | 78.56±0.95 | 87.56±0.53 | 86.13±0.64 |
| | $M^{S\text{DA}-\text{\beta}}$ (ours) | 72.82±1.13 | 98.43±0.68 | 96.14±0.81 | 81.32±0.86 | 89.58±0.56 | 87.65±0.75 |

Table 2. Digits Classification Results. $mt$, $up$, $sv$, $sy$, $mm$ are abbreviations for MNIST, USPS, SVHN, Synthetic Digits, MNIST-M, respectively. Our model $M^{S\text{DA}}$ and $M^{S\text{DA}-\text{\beta}}$ achieve 86.13% and 87.65% accuracy, outperforming other baselines by a large margin.

\(^3\text{http://pytorch.org}\)
Table 3. Single-source baselines on the LSDAC dataset. Several single-source adaptation baselines are evaluated on the LSDAC dataset, including AlexNet [18], DAN [25], JAN [27], DANN [8], RTN [26], ADDA [40], MCD [38], SE [7]. In each sub-table, the column-wise domains are selected as the source domain and the row-wise domains are selected as the target domain. The green numbers represent the average performance of each column or row. The red numbers denote the average accuracy for all the 30 (source, target) combinations.

Table 4. Results on Office-Caltech10 dataset. A,C,W and D represent Amazon, Caltech, Webcam and DSLR, respectively. All the experiments are based on ResNet-101 pre-trained on ImageNet.

5.3. Experiments on LSDAC

Single-Source Adaptation To demonstrate the intrinsic difficulty of LSDAC, we evaluate multiple state-of-the-art algorithms for single-source adaptation: Deep Alignment Network (DAN) [25], Joint Adaptation Network (JAN) [27], Domain Adversarial Neural Network (DANN) [8], Residual Transfer Network (RTN) [26], Adversarial Deep Domain Adaptation (ADDA) [40], Maximum Classifier Discrepancy (MCD) [38], and Self-Ensembling (SE) [7]. As the LSDAC dataset contains 6 domains, experiments for 30 different sources, target) combinations are performed for each baseline. For each domain, we follow a 70%/30% split scheme to participate our dataset.

Multi-Source Adaptation For both single best and source combine experiment setting, we take the following state-of-the-art methods as our baselines: Deep Alignment Network (DAN) [25], Joint Adaptation Network (JAN) [27], Domain Adversarial Neural Network (DANN) [8], Residual Transfer Network (RTN) [26], Adversarial Deep Domain Adaptation (ADDA) [40], Maximum Classifier Discrepancy (MCD) [38], and Self-Ensembling (SE) [7]. For multi-source domain adaptation, we take Deep Cocktail Network (DCTN) [45] as our baseline.

Results The experimental results of multi-source domain adaptation are shown in Table 5. We report the results of the two different weighting schemes and all the baseline results in Table 5. Our model M^S achieves an average accuracy of 41.5%, and M^S-β boosts the performance to 42.6%. The results demonstrate that our dataset.

Multi-Source Domain Adaptation LSDAC contains six domains. Inspired by Xu et al [45], we introduce two MSDA standards: (1) single best, reporting the single best-performing source transfer result on the test set, and (2) source combine, combining the source domains to a single domain and performing traditional single-source adaptation. The first standard evaluates whether MSDA can improve the best single source UDA results; the second testifies whether MSDA is necessary to exploit.

Baselines For both single best and source combine experiment setting, we take the following state-of-the-art methods as our baselines: Deep Alignment Network (DAN) [25], Joint Adaptation Network (JAN) [27], Domain Adversarial Neural Network (DANN) [8], Residual Transfer Network (RTN) [26], Adversarial Deep Domain Adaptation (ADDA) [40], Maximum Classifier Discrepancy (MCD) [38], and Self-Ensembling (SE) [7]. For multi-source domain adaptation, we take Deep Cocktail Network (DCTN) [45] as our baseline.

Results The experimental results of multi-source domain adaptation are shown in Table 5. We report the results of the two different weighting schemes and all the baseline results in Table 5. Our model M^S achieves an average accuracy of 41.5%, and M^S-β boosts the performance to 42.6%. The results demonstrate that our dataset.
models designed for MSDA outperform the single best UDA results, the source combine results, and the multi-source baseline. From the experimental results, we make three interesting observations. (1)The performance of M$^3$SDA$^*$ is 40.8%. After applying the weight vector $W$, M$^3$SDA$^*$ improves the mean accuracy by 0.7 percent. (2) In clp,inf,rel,skt $\rightarrow$ qdr setting, the performances of our models (as well as DCTN [45]) are worse than source-only baseline, which indicates that negative transfer [31] occurs. (3) In the source combine setting, the performances of DAN [25], RTN [26], JAN [27], DANN [8] are lower than the source only baseline, indicating the negative transfer happens when the training data are from multiple source domains.

### Effect of Category Number
To show how the number of categories affects the performance of state-of-the-art domain adaptation methods, we choose the painting $\rightarrow$ real setting in LSDAC and gradually increase the number of category from 20 to 345. The results are in Figure 4. An interesting observation is that when the number of categories is small (which is exactly the case in most domain adaptation benchmarks), all methods tend to perform well. However, their performances drop at different rates when the number of categories increases. For example, SE [7] performs the best when there is a limit number of categories, but worst when the number of categories is larger than 150.

### 5.4. Ablation Study
To show how much performance gain we can get through source-source alignment (S-S) and source-target (S-T) alignment, we perform ablation study based on our model. From Table 6, we observe the key factor to the performance boost is matching the moments of source distributions to the target distribution. Matching source domains with each other further boosts the performance. The experimental results empirically demonstrate that aligning source domains is essential for MSDA.

### 6. Conclusion
In this paper, we have collected, annotated and evaluated by far the largest domain adaptation dataset named LSDAC. The dataset is challenging due to the presence of notable domain gaps and a large number of categories. We hope it will be beneficial to evaluate future single- and multi-source UDA methods.

We have also proposed M$^3$SDA to align multiple source domains with the target domain. We derive a meaningful
error bound for our method under the framework of cross-moment divergence. Further, we incorporate the moment matching component into deep neural network and train the model in an end-to-end fashion. Extensive experiments on multi-source domain adaptation benchmarks demonstrate that our model outperforms all the multi-source baselines as well as the best single-source domain adaptation method.

7. Acknowledgement

We thank Ruiqi Gao, Yizhe Zhu, Saito Kuniaki, Ben Usman, Ping Hu for their useful discussions and suggestions. We thank anonymous annotators for their hard work to label the data. This work was partially supported by NSF and Honda Research Institute.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. Machine learning, 79(1-2):151–175, 2010. 1, 2, 3, 5, 12
[2] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, et al. Analysis of representations for domain adaptation methods. Advances in neural information processing systems, pages 137–144, 2007. 12
[3] K. Boussmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In Advances in Neural Information Processing Systems, pages 343–351, 2016. 2
[4] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. Journal of Machine Learning Research, 9(Aug):1757–1774, 2008. 1, 3, 12
[5] C. De Boor, C. De Boor, E.-U. Mathématien, C. De Boor, and C. De Boor. A practical guide to splines, volume 27. Springer-Verlag New York, 1978. 4
[6] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1338–1345. IEEE, 2012. 1, 3
[7] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In International Conference on Learning Representations, 2018. 2, 7, 8, 13, 14
[8] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. 1, 2, 6, 7, 8
[9] M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. In Pacific Rim international conference on artificial intelligence, pages 898–904. Springer, 2014. 2
[10] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In European Conference on Computer Vision, pages 597–613. Springer, 2016. 2
[11] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2066–2073. IEEE, 2012. 2, 6
[12] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In Advances in neural information processing systems, pages 513–520, 2007. 2
[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 7, 14
[14] J. Hoffman, M. Mohri, and N. Zhang. Algorithms and theory for multiple-source adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 8246–8256. Curran Associates, Inc., 2018. 1, 2, 3, 5
[15] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In J. Dy and A. Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 1899–1908, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 1, 3
[16] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In D. Precup and Y. W. Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1857–1865, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 3
[17] P. Koniusz, Y. Tas, H. Zhang, M. Harandi, F. Porikli, and R. Zhang. Museum exhibit identification challenge for the supervised domain adaptation and beyond. In The European Conference on Computer Vision (ECCV), September 2018. 2
[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 7
[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998. 2, 6
[20] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mind gan: Towards deeper understanding of moment matching network. In Advances in Neural Information Processing Systems, pages 2203–2213, 2017. 3
[21] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Deeper, broader and artier domain generalization. In International Conference on Computer Vision, 2017. 2
[22] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In International Conference on Machine Learning, pages 1718–1727, 2015. 3
[23] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In Advances in Neural Information Processing Systems, pages 700–708, 2017. 1
[24] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In Advances in neural information processing systems, pages 469–477, 2016.

[25] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 97–105, Lille, France, 07–09 Jul 2015. PMLR.

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In Advances in Neural Information Processing Systems, pages 136–144, 2016.

[27] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, pages 2220–2217, 2017.

[28] Y. Mansour, M. Mohri, A. Rostamizadeh, and A. R. Domain adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems 21, pages 1041–1048. Curran Associates, Inc., 2009.

[29] Y. Mroueh, T. Sercu, and V. Goel. McGan: Mean and covariation feature matching GAN. In D. Precup and Y. W. Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 2527–2535, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[30] O. Muradyan and S. Y. Khavinson. Absolute values of the coefficients of the polynomials in weierstrass’s approximation theorem. Mathematical notes of the Academy of Sciences of the USSR, 22(2):641–645, 1977.

[31] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 20(10):1345–1359, 2008.

[32] X. Peng and K. Saenko. Synthetic to real adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems 21, pages 1041–1048. Curran Associates, Inc., 2009.

[33] X. Peng, B. Sun, K. Ali, and K. Saenko. Learning deep object detectors from 3d models. In Proceedings of the IEEE International Conference on Computer Vision, pages 1278–1286, 2015.

[34] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924, 2017.

[35] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko. Syn2real: A new benchmark fprsynthetic-to-real visual domain adaptation. CoRR, abs/1806.09755, 2018.

[36] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. Dataset Shift in Machine Learning. The MIT Press, 2009.

[37] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In European conference on computer vision, pages 213–226. Springer, 2010.

[38] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[39] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In AAAI, volume 6, page 8, 2016.

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In Computer Vision and Pattern Recognition (CVPR), volume 1, page 4, 2017.

[41] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474, 2014.

[42] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In Measures of complexity, pages 11–30. Springer, 2015.

[43] H. Venkateswaran, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[44] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu. Visual domain adaptation with manifold embedded distribution alignment. In ACM Multimedia Conference, 2018.

[45] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3964–3973, 2018.

[46] Z. Yi, H. R. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In ICCV, pages 2868–2876, 2017.

[47] W. Zellinger, T. Grubinger, E. Lughhofer, T. Natschlager, and S. Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. CoRR, abs/1702.08811, 2017.

[48] Z. Zhang, M. Wang, Y. Huang, and A. Nehorai. Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3437–3445, 2018.

[49] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. In Advances in Neural Information Processing Systems, pages 8568–8579, 2018.

[50] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017.

[51] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He. Supervised representation learning: Transfer learning with deep autoencoders. In IJCAI, pages 4119–4125, 2015.
8. Appendix

The appendix is organized as follows: Section A introduces the formal definition of the cross-moment divergence; Section B gives the proof of Theorem 1 and further discussions; Section C provides the details of experiments on “Digit-Five” dataset; Section D shows feature visualization with t-SNE plot; Section E shows how the number of categories will affect the performance of the state-of-the-art models; Section F and Section G introduce the ResNet baselines and Train/Test split of our LSDAC dataset, respectively; Section H and Section I show the image samples and the detailed statistics of our LSDAC dataset.

A. Cross-moment Divergence

**Definition 2** (cross-moment divergence). Given a compact domain \( \mathcal{X} \subset \mathbb{R}^n \) and two probability measures \( \mu, \mu' \) on \( \mathcal{X} \), the \( k \)-th order cross-moment divergence between \( \mu \) and \( \mu' \) is

\[
d_{CM^k}(\mu, \mu') = \sum_{i \in \Delta_k} \left| \int_X \prod_{j=1}^n (x_j)^{i_j} d\mu(x) - \int_X \prod_{j=1}^n (x_j)^{i_j} d\mu'(x) \right|
\]

where \( \Delta_k = \{(i_1, i_2, \ldots, i_n) \in \mathbb{N}_0^n | \sum_{j=1}^n i_j = k \} \).

As seen in the rest of the paper, for two domains \( D = (\mu, f) \) and \( D' = (\mu', f') \), we use \( d_{CM^k}(D, D') \) to denote \( d_{CM^k}(\mu, \mu') \) for readability concerns.

B. Proof of Theorem 1

**Theorem 2** (Weierstrass Approximation Theorem). Let \( f : \mathcal{C} \rightarrow \mathbb{R} \) be continuous, where \( \mathcal{C} \) is a compact subset of \( \mathbb{R}^n \). There exists a sequence of real polynomials \( (P_m(x))_{m \in \mathbb{N}} \) such that

\[
\sup_{x \in \mathcal{C}} |f(x) - P_m(x)| \rightarrow 0, \quad \text{as} \quad m \rightarrow \infty.
\]

Note that for \( x \in \mathbb{R}^n \), a multivariate polynomial \( P_m : \mathbb{R}^n \rightarrow \mathbb{R} \) is of the form

\[
P_m(x) = \sum_{k=1}^m \sum_{i \in \Delta_k} a_i \prod_{j=1}^n (x_j)^{i_j},
\]

where \( \Delta_k = \{ (i_1, i_2, \ldots, i_n) \in \mathbb{N}_0^n | \sum_{j=1}^n i_j = k \} \).

**Lemma 3.** For any hypothesis \( h, h' \in \mathcal{H} \), for any \( \epsilon > 0 \), there exist an integer \( n_\epsilon \) and a constant \( a_{n_\epsilon} \), such that

\[
|\epsilon_S(h, h') - \epsilon_T(h, h')| \leq \frac{1}{2} a_{n_\epsilon} \sum_{k=1}^{n_\epsilon} d_{CM^k}(D_S, D_T) + \epsilon.
\]

**Proof.**

\[
|\epsilon_S(h, h') - \epsilon_T(h, h')| \leq \sup_{h, h' \in \mathcal{H}} |\epsilon_S(h, h') - \epsilon_T(h, h')| = \sup_{h, h' \in \mathcal{H}} \left| \mathbb{P}_{\mathcal{X} \sim D_S}[h(x) \neq h'(x)] - \mathbb{P}_{\mathcal{X} \sim D_T}[h(x) \neq h'(x)] \right|
\]

\[
= \sup_{h, h' \in \mathcal{H}} \left| \int_X 1_{h(x) \neq h'(x)} d\mu_S - \int_X 1_{h(x) \neq h'(x)} d\mu_T \right|,
\]

where \( \mathcal{X} \) is a compact subset of \( \mathbb{R}^n \). For any fixed \( h, h' \), the indicator function \( 1_{h(x) \neq h'(x)}(x) \) is a Lebesgue integrable function (\( L^1 \) function) on \( \mathcal{X} \). It is known that the space of continuous functions with compact support, denoted by \( \mathcal{C}_c(\mathcal{X}) \), is dense in \( L^1(\mathcal{X}) \), i.e., any \( L^1 \) function on \( \mathcal{X} \) can be approximated arbitrarily well\(^4\) by functions in \( \mathcal{C}_c(\mathcal{X}) \). As a result, for any \( \frac{\epsilon}{2} > 0 \), there exists \( f \in \mathcal{C}_c(\mathcal{X}) \), such that,

\[
\sup_{h, h' \in \mathcal{H}} \left| \int_X 1_{h(x) \neq h'(x)} d\mu_S - \int_X 1_{h(x) \neq h'(x)} d\mu_T \right| \leq \int_X |f(x)| d\mu_S - \int_X |f(x)| d\mu_T + \frac{\epsilon}{2}.
\]

**Proof.**

\[
|\epsilon_S(h, h') - \epsilon_T(h, h')| \leq \sup_{h, h' \in \mathcal{H}} |\epsilon_S(h, h') - \epsilon_T(h, h')| = \sup_{h, h' \in \mathcal{H}} \left| \mathbb{P}_{\mathcal{X} \sim D_S}[h(x) \neq h'(x)] - \mathbb{P}_{\mathcal{X} \sim D_T}[h(x) \neq h'(x)] \right|
\]

\[
= \sup_{h, h' \in \mathcal{H}} \left| \int_X 1_{h(x) \neq h'(x)} d\mu_S - \int_X 1_{h(x) \neq h'(x)} d\mu_T \right|,
\]

where \( \mathcal{X} \) is a compact subset of \( \mathbb{R}^n \). For any fixed \( h, h' \), the indicator function \( 1_{h(x) \neq h'(x)}(x) \) is a Lebesgue integrable function (\( L^1 \) function) on \( \mathcal{X} \). It is known that the space of continuous functions with compact support, denoted by \( \mathcal{C}_c(\mathcal{X}) \), is dense in \( L^1(\mathcal{X}) \), i.e., any \( L^1 \) function on \( \mathcal{X} \) can be approximated arbitrarily well\(^4\) by functions in \( \mathcal{C}_c(\mathcal{X}) \). As a result, for any \( \frac{\epsilon}{2} > 0 \), there exists \( f \in \mathcal{C}_c(\mathcal{X}) \), such that,

\[
\sup_{h, h' \in \mathcal{H}} \left| \int_X 1_{h(x) \neq h'(x)} d\mu_S - \int_X 1_{h(x) \neq h'(x)} d\mu_T \right| \leq \int_X |f(x)| d\mu_S - \int_X |f(x)| d\mu_T + \frac{\epsilon}{2}.
\]

\(^4\)with respect to the corresponding norm.
\[ P_n(x) = \sum_{k=1}^{n_k} \sum_{i \in \Delta_k} \alpha_i \prod_{j=1}^{n_i} (x_j)^{t_j}, \text{ such that} \]

\[
\left| \int_X f(x) d\mu_S - \int_X f(x) d\mu_T \right| \\
\leq \int_X P_n(x) d\mu_S - \int_X P_n(x) d\mu_T + \frac{\epsilon}{2} \\
\leq \sum_{k=1}^{n_k} \left| \sum_{i \in \Delta_k} \alpha_i \int_X \prod_{j=1}^{n_i} (x_j)^{t_j} d\mu_S \right| \\
- \sum_{i \in \Delta_k} \alpha_i \int_X \prod_{j=1}^{n_i} (x_j)^{t_j} d\mu_T + \frac{\epsilon}{2} \\
\leq \sum_{k=1}^{n_k} \sum_{i \in \Delta_k} \left( |a_i| \int_X \prod_{j=1}^{n_i} (x_j)^{t_j} d\mu_S \right) \\
- \sum_{k=1}^{n_k} \sum_{i \in \Delta_k} \left| \int_X \prod_{j=1}^{n_i} (x_j)^{t_j} d\mu_T \right| + \frac{\epsilon}{2} \\
\leq \sum_{k=1}^{n_k} \sum_{i \in \Delta_k} a_{\Delta_k} \sum_{i \in \Delta_k} \left( \int_X \prod_{j=1}^{n_i} (x_j)^{t_j} d\mu_S \right) \\
- \sum_{i \in \Delta_k} \alpha_i \int_X \prod_{j=1}^{n_i} (x_j)^{t_j} d\mu_T + \frac{\epsilon}{2} \\
= \sum_{k=1}^{n_k} a_{\Delta_k} d_{CM}(D_S, D_T) + \frac{\epsilon}{2} \\
\leq \frac{1}{2} \sum_{k=1}^{n_k} a_{n_k} d_{CM}(D_S, D_T) + \frac{\epsilon}{2}, \quad (9)
\]

where \( a_{\Delta_k} = \max_{i \in \Delta_k} |a_i| \) and \( a_{n_k} = 2 \max_{1 \leq k \leq n_k} a_{\Delta_k} \). Combining Equation 7, 8, 9, we prove the lemma. \( \square \)

Note that the constants \( a_{\Delta_k} \) can actually be meaningfully bounded when applying the Weierstrass Approximation Theorem. According to [30], a sequence of positive numbers \( \{M_k\} \) can be constructed, such that, for any \( \epsilon > 0 \), there exists a polynomial \( P_{n_k} \) such that \( |P_{n_k}(x) - f(x)| < \epsilon \) and \( |a_{\Delta_k}| < \epsilon M_k, \forall k = 1, \ldots, n_k \).

**Lemma 4 (Lemma 6, [1]).** For each \( D_j \in \{D_1, \ldots, D_N\} \), let \( S_j \) be a labeled sample set of size \( \beta_j m \) drawn from \( \mu_j \) and labeled by the groundtruth labeling function \( f_j \). For any fixed weight vector \( \alpha \), let \( \hat{\epsilon}(h) \) be the empirical \( \alpha \)-weighted error of some fixed hypothesis \( h \) on these sample sets, and let \( \epsilon(h) \) be the true \( \alpha \)-weighted error, then

\[ P[|\hat{\epsilon}(h) - \epsilon(h)| \geq \epsilon] \leq 2 \exp \left( \frac{-2m\epsilon^2}{\sum_{j=1}^{N} \alpha_j^2} \right). \]

**Lemma 4** is a slight modification of the Hoeffdings inequality for the empirical \( \alpha \)-weighted source error, which will be useful in proving the uniform convergence bound for hypothesis space of finite VC dimension. Now we are ready to prove Theorem 1.

**Theorem 1.** Let \( \mathcal{H} \) be a hypothesis space of VC dimension \( d \). Let \( n \) be the size of labeled samples from all sources \( \{D_1, D_2, \ldots, D_N\} \). \( S_j \) be the labeled sample set of size \( \beta_j m \) \( (\sum_j \beta_j = 1) \) drawn from \( \mu_j \) and labeled by the groundtruth labeling function \( f_j \). If \( h \in \mathcal{H} \) is the empirical minimizer of \( \hat{\epsilon}(\alpha)(h) \) for a fixed weight vector \( \alpha \) and \( h^* = \min_{h \in \mathcal{H}} \epsilon_T(h) \) is the target error minimizer, then for any \( \delta \in (0, 1) \) and any \( \epsilon > 0 \), there exist \( N \) integers \( \{n_j^i\}_{j=1}^{N} \) and \( N \) constants \( \{\alpha_j^i\}_{j=1}^{N} \), such that with probability at least \( 1 - \delta \),

\[ \epsilon_T(h) \leq \epsilon_T(h^*) + \eta_{\alpha, \beta, m, \delta} + \epsilon \]
\[ + \sum_{j=1}^{N} \alpha_j \left( 2\lambda_j + a_{n_j^i} \sum_{k=1}^{n_j^i} d_{CM}(D_j, D_T) \right), \quad (6) \]

where \( \eta_{\alpha, \beta, m, \delta} = 4 \sqrt{\left( \sum_{j=1}^{N} \frac{\alpha_j^2}{\lambda_j^2} \right)(2d(\log(\frac{2m}{\delta})+1)+2 \log(\frac{1}{\delta}))} \)

and \( \lambda_j = \min_{h \in \mathcal{H}} \left( \epsilon_T(h) + \epsilon(h) \right) \).

**Proof.** Let \( h_j^* = \arg \min_{h \in \mathcal{H}} (\epsilon_T(h) + \epsilon(h)) \). Then for any \( \epsilon > 0 \), there exists \( N \) integers \( \{n_j^i\}_{j=1}^{N} \) and \( N \) constant \( \{\alpha_j^i\}_{j=1}^{N} \), such that

\[ |\epsilon(h) - \epsilon_T(h)| \leq \left| \sum_{j=1}^{N} \alpha_j \epsilon(h_j) - \epsilon_T(h) \right| \leq \sum_{j=1}^{N} \alpha_j |\epsilon(h_j) - \epsilon_T(h)| \]
\[ \leq \sum_{j=1}^{N} \alpha_j (|\epsilon(h_j) - \epsilon_j(h_j, h_j^*)| + |\epsilon_j(h_j, h_j^*) - \epsilon_T(h_j, h_j^*)|) \]
\[ + \left| \epsilon_T(h_j, h_j^*) - \epsilon_T(h) \right| \]
\[ \leq \sum_{j=1}^{N} \alpha_j (\epsilon_j(h_j^*) + |\epsilon_j(h_j, h_j^*) - \epsilon_T(h_j, h_j^*)|) \]
\[ \leq \sum_{j=1}^{N} \alpha_j \left( \lambda_j + \frac{1}{2} a_{n_j^i} \sum_{k=1}^{n_j^i} d_{CM}(D_j, D_T) \right) + \frac{\epsilon}{2}. \quad (10) \]

The third inequality follows from the triangle inequality of classification error\(^5\) [2, 4]. The last inequality follows from the definition of \( \lambda_j \) and Lemma 3. Now using both Equation 10 and Lemma 4, we have for any \( \delta \in (0, 1) \) and any

---

\(^5\)For any labeling function \( f_1, f_2, f_3 \), we have \( \epsilon(f_1, f_2) \leq \epsilon(f_1, f_3) + \epsilon(f_2, f_3) \).
\[
\epsilon_T(h) \leq \epsilon_{\alpha}(\hat{h}) + \frac{\epsilon}{2} + \sum_{j=1}^{N} \alpha_j \left( \lambda_j + \frac{1}{2} a_{\omega_j}^{2} \sum_{k=1}^{n_j} d_{CM}(D_j, D_T) \right) \\
\leq \epsilon_{\alpha}(\hat{h}) + \frac{1}{2} \eta_{\alpha, \beta, m, \delta} + \frac{\epsilon}{2} + \sum_{j=1}^{N} \alpha_j \left( \lambda_j + \frac{1}{2} a_{\omega_j}^{2} \sum_{k=1}^{n_j} d_{CM}(D_j, D_T) \right) \\
\leq \epsilon_{\alpha}(\hat{h}^*) + \frac{1}{2} \eta_{\alpha, \beta, m, \delta} + \frac{\epsilon}{2} + \sum_{j=1}^{N} \alpha_j \left( \lambda_j + \frac{1}{2} a_{\omega_j}^{2} \sum_{k=1}^{n_j} d_{CM}(D_j, D_T) \right) \\
\leq \epsilon_{\alpha}(h^*) + \frac{1}{2} \eta_{\alpha, \beta, m, \delta} + \frac{\epsilon}{2} + \sum_{j=1}^{N} \alpha_j \left( 2\lambda_j + \frac{1}{2} a_{\omega_j}^{2} \sum_{k=1}^{n_j} d_{CM}(D_j, D_T) \right).
\]

The first and the last inequalities follow from Equation 10, the second and the fourth inequalities follow from applying Lemma 4 (instead of standard Hoeffding’s inequality) in the standard proof of uniform convergence for empirical risk minimizers [42]. The third inequality follows from the definition of \( \hat{h} \).

To better understand the bounds in Theorem 1, the second term of the bound is the VC-dimension based generalization error, which is the upper bound of the difference between the empirical error \( \epsilon_{\alpha} \) and the true expected error \( \epsilon_T \). The last term (a summation), as shown in Equation 10, characterizes the upper bound of the difference between the \( \alpha \)-weighted error \( \epsilon_{\alpha} \) and the target error \( \epsilon_T \). The constants \( \{a_{\omega_j}^{2}\}_{j=1}^{N} \) in this term can be meaningfully bounded, as explained at the end of the proof of Lemma 3.

Note that the bound explicitly depends on cross-moment divergence terms \( d_{CM}(D_j, D_T) \), and thus sheds new light on the theoretical motivation of moment matching approaches, including our proposed approach and many existing approaches for both single and multiple source domain adaptation. To the best of our knowledge, this is the first target error bound in the literature of domain adaptation that explicitly incorporates a moment-based divergence between the source(s) and the target domains.

C. Details of Digit Experiments

Network Architecture In our digital experiments (Table 2), our feature extractor is composed of three \( \text{conv} \) layers and two \( \text{fc} \) layers. We present the \( \text{conv} \) layers as \( \text{(input, output, kernel, stride, padding)} \) and \( \text{fc} \) layers as \( \text{(input, output)} \). The three \( \text{conv} \) layers are: \( \text{conv1} \) (3, 64, 5, 1, 2), \( \text{conv2} \) (64, 64, 5, 1, 2), \( \text{conv3} \) (64, 128, 5, 1, 2). The two \( \text{fc} \) layers are: \( \text{fc1} \) (8192, 3072), \( \text{fc2} \) (3072, 2048). The architecture of the feature generator is: (\( \text{conv} \)1, \( bn1, relu1, pool1 \))-(\( \text{conv} \)2, \( bn2, relu2, pool2 \))-(\( \text{conv} \)3, \( bn3, relu3 \))-(\( fc1, bn4, relu4, dropout \))-(\( fc2, bn5, relu5 \)). The classifier is a single \( fc \) layer, i.e. \( fc3 \) (2048, 10).

Convergence Analysis As our framework involves multiple classifiers and the model is trained with multiple losses, we visualize the learning procedure of \( mn, mt, sv, sy \rightarrow up \) setting in Figure 7. The figure shows the training errors of multiple classifiers are decreasing, despite some frequent deviations. The MD (Moment Discrepancy) loss decreases in a steady way, demonstrating that the MD loss and the cross-entropy loss gradually converge.

D. Feature visualization

To demonstrate the transfer ability of our model, we visualize the DAN [25] features and \( M^3 \text{SDA-} \beta \) features with t-SNE embedding in two tasks: \( mn, mt, sv, sy \rightarrow up \) and A.D.W \( \rightarrow \) C. The results are shown in Figure 6. We make two important observations: i) Comparing Figure 6(a) with Figure 6(b), we find that \( M^3 \text{SDA-} \beta \) is capable of learning more discriminative features; ii) From Figure 6(c) and Figure 6(d), we find that the clusters of \( M^3 \text{SDA-} \beta \) features are more compact than those of DAN, which suggests that the features learned by \( M^3 \text{SDA-} \beta \) attain more desirable discriminative property. These observations imply the superiority of our model over DAN in multi-source domain adaptation.

E. Effect of Category Number

In this section, we show more results to clarify how the number of categories affects the performances of the state-of-the-art models. We choose the following four settings, i.e., \( \text{painting} \rightarrow \text{real} \) (Figure 5(a)), \( \text{infograph} \rightarrow \text{real} \) (Figure 5(b)), \( \text{sketch} \rightarrow \text{clipart} \) (Figure 5(c)), \( \text{quickdraw} \rightarrow \text{clipart} \) (Figure 5(d)), and gradually increase the number of categories from 20 to 345. From the four figures, we make the following interesting observations:

- All the models perform well when the number of categories is small. However, their performances drop rapidly when the number of categories increases.
- Self-Ensembling [7] model has a good performance when the number of categories is small, but it is not suitable for large scale domain adaptation.
Table 7. **Single-source ResNet101 and ResNet152** [13] baselines on the LSDAC dataset. We provide ResNet baselines for Table 3. In each sub-table, the column-wise domains are selected as the source domain and the row-wise domains are selected as the target domain. The green numbers represent the average performance of each column or row. The red numbers denote the average accuracy for all the 30 (source, target) combinations. *(clp: clipart, inf: infograph, pnt: painting, qdr: quickdraw, rel: real, skt: sketch).*

|          | clp     | inf     | pnt     | qdr     | rel     | skt     | Avg.  |
|----------|---------|---------|---------|---------|---------|---------|-------|
| ResNet101| clp     | inf     | pnt     | qdr     | rel     | skt     | Avg.  |
|          | N/A     | 19.3%   | 37.5%   | 11.1%   | 52.2%   | 41.0%   | 32.2% |
|          | 30.2%   | N/A     | 31.2%   | 3.6%    | 44.0%   | 27.9%   | 27.4% |
| Avg.     | 3.4%    | 15.3%   | 31.3%   | 7.4%    | 40.4%   | 30.5%   | 26.5% |

*Table 7: Single-source ResNet101 and ResNet152 [13] baselines on the LSDAC dataset.* We provide ResNet baselines for Table 3. In each sub-table, the column-wise domains are selected as the source domain and the row-wise domains are selected as the target domain. The green numbers represent the average performance of each column or row. The red numbers denote the average accuracy for all the 30 (source, target) combinations. *(clp: clipart, inf: infograph, pnt: painting, qdr: quickdraw, rel: real, skt: sketch.)*

**Figure 5. Accuracy vs. Number of Categories.** We plot how the performances of different models will change when the number of categories increases. We select four UDA settings, i.e., `painting→real`, `infograph→real`, `sketch→clipart`, `quickdraw→clipart`. The figure shows the performance of all models drop significantly with the increase of category number.

**F. ResNet baselines**

We report ResNet [13] source only baselines in Table 7. The MCD [38] and the SE [7] methods (In Table 3) are based on ResNet101 and ResNet152, respectively. We have observed these two methods both perform worse than their source only baselines, which indicates negative transfer [31] phenomenon occurs in these two scenarios. Exploring why negative transfer happens is beyond the scope of this literature. One preliminary guess is due to the large number of categories.

**G. Train/Test Split**

We show the detailed number of images we used in our experiments in Table 8. For each domain, we follow a 70%/30% schema to split the dataset to training and testing trunk. The “Per-Class” row shows the average number of images that each category contains.
Figure 6. Feature visualization: t-SNE plot of DAN features and M\(^3\)SDA-$\beta$ features on SVHN in mm,mt,up,sv setting; t-SNE of DAN features and M\(^3\)SDA-$\beta$ features on Caltech in A,D,W→C setting. We use different markers and different colors to denote different categories. (Best viewed in color.)

Figure 7. Analysis: training error of each classifier, Moment Discrepancy (MD, defined in Equation 1), and accuracy (red bold line) w.r.t. training epochs in mm,mt,sv,sy→up setting. The bold red line shows how the accuracy changes w.r.t. training epochs.

H. Image Samples

We sample the images for each domain and show them in Figure 8 (clipart), Figure 9 (infograph), Figure 10 (painting), Figure 11 (quickdraw), Figure 12 (real), and Figure 13 (sketch).

I. Dataset Statistics

Table 9, Table 10, and Table 11 show the detailed statistics of our LSDAC dataset. Our dataset contains 6 distinct domains, 345 categories and ~0.6 million images. The categories are from 24 divisions, which are: Furniture, Mammal, Tool, Cloth, Electricity, Building, Office, Human Body, Road Transportation, Food, Nature, Cold Blooded, Music, Fruit, Sport, Tree, Bird, Vegetable, Shape, Kitchen, Water Transportation, Sky Transportation, Insect, Others.
Figure 8. Images sampled from clipart domain of the LSDAC dataset.
Figure 9. Images sampled from *infograph* domain of the LSDAC dataset.
Figure 10. Images sampled from painting domain of the LSDAC dataset.
Figure 11. Images sampled from quickdraw domain of the LSDAC dataset.
Figure 12. Images sampled from real domain of the LSDAC dataset.
Figure 13. Images sampled from sketch domain of the LSDAC dataset.
| class     | clp  | inf  | put  | qdr  | rel  | skt  | total   |
|-----------|------|------|------|------|------|------|---------|
| bathtub   | 100  | 135  | 500  | 517  | 210  | 500  | 1507    |
| ceiling fan | 35   | 63   | 38   | 500  | 217  | 25   | 878     |
| couch     | 232  | 61   | 250  | 601  | 60   | 1400 |         |
| fence     | 165  | 99   | 49   | 500  | 770  | 140  | 1723    |
| hot tub   | 144  | 86   | 197  | 500  | 757  | 49   |         |
| mailbox   | 18   | 45   | 101  | 500  | 595  | 151  | 1410    |
| postcard  | 91   | 37   | 88   | 500  | 888  | 1401 |         |
| sleeping bag | 96  | 14   | 17   | 500  | 406  | 591  | 1624    |
| streetlight | 326  | 113  | 537  | 500  | 463  | 268  | 2207    |
| table     | 297  | 736  | 104  | 500  | 563  | 2500 |         |
| toothbrush | 159  | 556  | 11   | 500  | 582  | 235  | 2043    |
| vase      | 161  | 319  | 262  | 500  | 632  | 187  | 2061    |
| bear      | 124  | 81   | 379  | 500  | 585  | 178  | 1847    |
| cow       | 188  | 134  | 156  | 500  | 541  | 17   | 1536    |
| elephant  | 115  | 188  | 425  | 500  | 789  | 266  | 2283    |
| horse     | 201  | 216  | 521  | 500  | 645  | 103  | 2186    |
| monkey    | 123  | 85   | 405  | 500  | 699  | 166  | 1978    |
| pig       | 93   | 203  | 326  | 500  | 577  | 227  | 1926    |
| rhinoceros | 102  | 91   | 220  | 500  | 684  | 183  | 1780    |
| tiger     | 315  | 285  | 422  | 500  | 607  | 386  | 2515    |
| wood      | 48   | 92   | 219  | 500  | 382  | 219  | 1460    |
| axe       | 92   | 45   | 41   | 500  | 628  | 120  | 1426    |
| bucket    | 142  | 56   | 206  | 500  | 335  | 152  | 1663    |
| dumbbell  | 387  | 86   | 189  | 500  | 581  | 190  | 1933    |
| nail      | 41   | 256  | 838  | 500  | 674  | 23   | 2332    |
| pliers    | 38   | 65   | 293  | 500  | 453  | 163  | 1512    |
| saw       | 76   | 34   | 150  | 500  | 118  | 110  | 988     |
| skateboard | 263  | 50   | 152  | 500  | 577  | 419  | 1941    |
| sword     | 139  | 124  | 470  | 500  | 591  | 384  | 2208    |
| bracelet  | 293  | 123  | 150  | 500  | 715  | 300  | 2081    |
| crown     | 208  | 17   | 81   | 500  | 170  | 1152 | 1152    |
| jacket    | 72   | 82   | 272  | 500  | 457  | 84   | 1467    |
| pants     | 16   | 173  | 381  | 500  | 398  | 136  | 1604    |
| shoe      | 127  | 291  | 500  | 587  | 645  | 2410  |
| sweater   | 222  | 92   | 153  | 500  | 579  | 167  | 1713    |
| t-shirt   | 98   | 320  | 12   | 500  | 537  | 220  | 1652    |
| camera    | 58   | 66   | 156  | 500  | 480  | 109  | 1369    |
| cooler    | 214  | 21   | 13   | 500  | 528  | 90   | 1366    |
| flash light | 221  | 62   | 418  | 500  | 461  | 95   | 1575    |
| laptop    | 26   | 118  | 161  | 500  | 387  | 319  | 1511    |
| microphone | 143  | 70   | 152  | 500  | 562  | 156  | 1583    |
| power outlet | 25  | 76   | 102  | 500  | 620  | 95   | 1418    |
| spreadsheet | 187  | 397  | 34  | 500  | 751  | 677  | 2546    |
| television | 136  | 546  | 51  | 500  | 400  | 127  | 1760    |
| cell phone | 38   | 170  | 136  | 500  | 520  | 23   | 1387    |
| dishwasher | 109  | 47   | 107  | 500  | 508  | 40   | 1311    |
| headphones | 285  | 224  | 181  | 500  | 551  | 188  | 1929    |
| light bulb | 12   | 185  | 482  | 500  | 262  | 405  | 1846    |
| microwave | 16   | 114  | 10   | 500  | 338  | 170  | 1148    |
| radio     | 30   | 101  | 65   | 500  | 398  | 165  | 1259    |
| toaster   | 289  | 334  | 12   | 500  | 211  | 90   | 1436    |

Table 9. Detailed statistics of the LSDAC dataset.
## Building

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| The Eiffel Tower | 114  | 190  | 321  | 500  | 553  | 276  | 1954   |
| bridge           | 66   | 61   | 471  | 500  | 769  | 335  | 2202   |
| diving board     | 182  | 127  | 500  | 593  | 71   | 1485  |
| golf club        | 207  | 169  | 650  | 500  | 552  | 695  | 2773   |
| jail             | 104  | 26   | 54   | 500  | 587  | 94   | 1365   |
| pool             | 139  | 173  | 90   | 500  | 680  | 103  | 1685   |
| tent             | 153  | 234  | 141  | 500  | 590  | 339  | 1957   |

## Human Body

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| arm              | 50   | 129  | 422  | 500  | 235  | 1585 |        |
| ear              | 101  | 58   | 187  | 500  | 348  | 199  | 1393   |
| face             | 54   | 110  | 20   | 500  | 696  | 452  | 1832   |
| goatee           | 255  | 236  | 129  | 500  | 562  | 219  | 1901   |
| leg              | 89   | 174  | 178  | 500  | 659  | 145  | 1745   |
| nose             | 57   | 226  | 512  | 500  | 362  | 103  | 1760   |
| toe              | 85   | 407  | 12   | 500  | 356  | 78   | 1428   |

## Road Transportation

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| ambulance        | 80   | 20   | 74   | 500  | 623  | 115  | 1412   |
| bus              | 101  | 183  | 112  | 500  | 745  | 233  | 1874   |
| motorbike        | 42   | 209  | 106  | 500  | 772  | 209  | 1838   |
| roller coaster    | 143  | 46   | 75   | 500  | 637  | 61   | 1462   |
| train            | 109  | 373  | 406  | 500  | 240  | 2309  |

## Nature

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| beach            | 105  | 183  | 499  | 500  | 622  | 79   | 1988   |
| lightning        | 171  | 68   | 199  | 500  | 560  | 94   | 1592   |
| ocean            | 54   | 47   | 475  | 500  | 591  | 77   | 1744   |
| river            | 134  | 155  | 558  | 500  | 651  | 111  | 2109   |
| sun              | 248  | 352  | 572  | 500  | 161  | 258  | 2091   |

## Cold Blooded

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| crab             | 108  | 50   | 153  | 500  | 717  | 152  | 1680   |
| frog             | 163  | 118  | 167  | 500  | 761  | 203  | 1912   |
| scorpion         | 171  | 53   | 133  | 500  | 447  | 155  | 1759   |
| snail            | 166  | 18   | 321  | 500  | 465  | 405  | 1875   |

## Music

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| cello            | 93   | 34   | 158  | 500  | 450  | 64   | 1299   |
| guitar           | 103  | 204  | 203  | 500  | 632  | 183  | 1825   |
| saxophone        | 236  | 74   | 358  | 500  | 482  | 310  | 1960   |
| violin           | 174  | 282  | 203  | 500  | 512  | 203  | 1874   |

| class            | clp  | inf  | pnt  | qdr  | rel  | skt  | total  |
|------------------|------|------|------|------|------|------|--------|
| clarinet         | 214  | 25   | 89   | 500  | 407  | 33   | 1268   |
| guitar           | 258  | 37   | 224  | 500  | 649  | 45   | 1713   |
| saxophone        | 227  | 195  | 175  | 500  | 484  | 191  | 1772   |
| violin           | 117  | 247  | 122  | 500  | 391  | 188  | 1565   |

Table 10. Detailed statistics of the LSDAC dataset.
| Class       | clp | inf | pnt | qdr | rel | skt | total |
|-------------|-----|-----|-----|-----|-----|-----|-------|
| **Fruit**   |     |     |     |     |     |     |       |
| apple       | 88  | 75  | 445 | 500 | 54  | 181 | 1343  |
| blueberry   | 171 | 110 | 167 | 500 | 733 | 129 | 1810  |
| pineapple   | 83  | 133 | 333 | 500 | 673 | 131 | 1859  |
| banana      | 50  | 376 | 159 | 500 | 258 | 204 | 1747  |
| strawberry  | 357 | 308 | 530 | 500 | 454 | 198 | 2347  |
| blackberry  | 106 | 214 | 500 | 568 | 60  | 1462 |
| pear        | 74  | 115 | 448 | 500 | 438 | 183 | 1758  |
| watermelon  | 193 | 401 | 500 | 671 | 128 | 2303 |
| **Sport**   |     |     |     |     |     |     |       |
| baseball    | 116 | 369 | 122 | 500 | 87  | 48  | 1242  |
| baseball bat| 106 | 353 | 145 | 500 | 118 | 196 | 1418  |
| hockey puck | 188 | 59  | 236 | 500 | 400 | 95  | 1478  |
| soccer ball | 85  | 163 | 268 | 500 | 272 | 377 | 1665  |
| basketball  | 61  | 219 | 276 | 500 | 237 | 160 | 1453  |
| hockey stick| 155 | 197 | 194 | 500 | 394 | 110 | 1559  |
| tennis racquet| 187 | 195 | 18  | 500 | 456 | 202 | 1558  |
| **Tree**    |     |     |     |     |     |     |       |
| bush        | 46  | 12  | 67  | 500 | 31  | 626 | 1282  |
| cactus      | 119 | 36  | 122 | 500 | 658 | 61  | 1496  |
| house plant | 25  | 292 | 416 | 500 | 484 | 156 | 1747  |
| leaf        | 253 | 140 | 485 | 500 | 360 | 336 | 2074  |
| **Bird**    |     |     |     |     |     |     |       |
| bird        | 336 | 208 | 222 | 500 | 803 | 306 | 2375  |
| owl         | 133 | 114 | 496 | 500 | 757 | 202 | 2202  |
| flamingo    | 274 | 294 | 500 | 542 | 142 | 1721 |
| **Vegetable**|  |     |     |     |     |     |       |
| asparagus   | 49  | 134 | 408 | 500 | 659 | 209 | 1595  |
| mushroom    | 136 | 298 | 254 | 500 | 788 | 252 | 2228  |
| potato      | 86  | 61  | 58  | 500 | 608 | 83  | 1396  |
| carrot      | 52  | 251 | 419 | 500 | 404 | 276 | 1792  |
| **Shape**   |     |     |     |     |     |     |       |
| circle      | 199 | 248 | 292 | 500 | 259 | 202 | 1700  |
| hexagon     | 196 | 362 | 160 | 500 | 592 | 116 | 1926  |
| line        | 13  | 210 | 502 | 500 | 102 | 25  | 1352  |
| **Kitchen** |     |     |     |     |     |     |       |
| fork        | 200 | 63  | 84  | 500 | 351 | 176 | 1374  |
| knife       | 32  | 108 | 30  | 500 | 582 | 129 | 1381  |
| spoon       | 228 | 127 | 158 | 500 | 534 | 406 | 1953  |
| hourglass   | 100 | 100 | 206 | 500 | 289 | 134 | 1329  |
| **Water Transportation** | |     |     |     |     |     |       |
| aircraft carrier | 27  | 88  | 133 | 500 | 390 | 63  | 1201  |
| sailboat    | 162 | 119 | 322 | 500 | 422 | 361 | 1886  |
| canoe       | 68  | 71  | 395 | 500 | 703 | 129 | 1866  |
| speedboat   | 271 | 76  | 141 | 500 | 620 | 487 | 2095  |
| submarine   | 344 | 183 | 550 | 500 | 607 | 207 | 2391  |
| **Sky Transportation** | |     |     |     |     |     |       |
| airplane    | 73  | 62  | 212 | 500 | 218 | 331 | 1396  |
| parachute  | 82  | 60  | 140 | 500 | 629 | 233 | 1644  |
| helicopter | 145 | 216 | 257 | 500 | 804 | 200 | 2122  |
| **Insect**  |     |     |     |     |     |     |       |
| ant         | 81  | 62  | 235 | 500 | 381 | 111 | 1370  |
| mosquito   | 56  | 232 | 65  | 500 | 562 | 144 | 1559  |
| bee         | 202 | 233 | 313 | 500 | 452 | 184| 2344  |
| butterfly  | 160 | 162 | 387 | 500 | 658 | 249 | 2116  |
| **Others**  |     |     |     |     |     |     |       |
| The Mona Lisa | 150 | 112 | 191 | 500 | 289 | 145 | 1387  |
| campfire   | 122 | 53  | 217 | 500 | 489 | 86  | 1467  |
| feather    | 268 | 432 | 344 | 500 | 505 | 336 | 2385  |
| snowman    | 174 | 123 | 901 | 500 | 114 | 712 | 2524  |
| traffic light | 211 | 280 | 60  | 500 | 379 | 127 | 1557  |

Table 11. Detailed statistics of the LSDAC dataset.