Multi-source Domain Adaptation in the Deep Learning Era: A Systematic Survey

Sicheng Zhao\textsuperscript{1}, Bo Li\textsuperscript{1}, Colorado Reed\textsuperscript{1}, Pengfei Xu\textsuperscript{2}, Kurt Keutzer\textsuperscript{1}

\textsuperscript{1}University of California, Berkeley, USA \textsuperscript{2}Didi Chuxing, China

schzhao@gmail.com, drluodian@gmail.com, cjrd@cs.berkeley.edu, xupengfeipf@didiglobal.com, keutzer@berkeley.edu

Abstract

In many practical applications, it is often difficult and expensive to obtain enough large-scale labeled data to train deep neural networks to their full capability. Therefore, transferring the learned knowledge from a separate, labeled source domain to an unlabeled or sparsely labeled target domain becomes an appealing alternative. However, direct transfer often results in significant performance decay due to domain shift. Domain adaptation (DA) addresses this problem by minimizing the impact of domain shift between the source and target domains. Multi-source domain adaptation (MDA) is a powerful extension in which the labeled data may be collected from multiple sources with different distributions. Due to the success of DA methods and the prevalence of multi-source data, MDA has attracted increasing attention in both academia and industry. In this survey, we define various MDA strategies and summarize available datasets for evaluation. We also compare modern MDA methods in the deep learning era, including latent space transformation and intermediate domain generation. Finally, we discuss future research directions for MDA.

1 Background and Motivation

The availability of large-scale labeled training data, such as ImageNet, has enabled deep neural networks (DNNs) to achieve remarkable success in many learning tasks, ranging from computer vision to natural language processing. For example, the classification error of the “Classification + localization with provided training data” task in the Large Scale Visual Recognition Challenge has reduced from 0.28 in 2010 to 0.0225 in 2017\textsuperscript{1}, outperforming even human classification. However, in many practical applications, obtaining labeled training data is often expensive, time-consuming, or even impossible. For example, in fine-grained recognition, the experts can provide reliable labels [Gebru et al., 2017]; in semantic segmentation, it takes about 90 minutes to label each Cityscapes image [Cordts et al., 2016]; in autonomous driving, it is difficult to label point-wise 3D LiDAR point clouds [Wu et al., 2019].

One potential solution is to transfer a model trained on a separate, labeled source domain to the desired unlabeled or sparsely labeled target domain. But as Figure 1 demonstrates, the direct transfer of models across domains leads to poor performance. Figure 1(a) shows that even for the simple task of digit recognition, training on the MNIST source domain [LeCun et al., 1998] for digit classification in the MNIST-M target domain [Ganin and Lempitsky, 2015] leads to a digit classification accuracy decrease from 96.0% to 52.3% when training a LeNet-5 model [LeCun et al., 1998]. Figure 1(b) shows a more realistic example of training a semantic segmentation model on a synthetic source dataset GTA [Richter et al., 2016] and conducting pixel-wise segmentation on a real target dataset Cityscapes [Cordts et al., 2016] using the FCN model [Long et al., 2015a]. If we train on the real data, we obtain a mean intersection-over-union (mIoU) of 62.6%; but if we train on synthetic data, the mIoU drops significantly to 21.7%.

The poor performance from directly transferring models across domains stems from a phenomenon known as domain shift [Torralba and Efros, 2011; Zhao et al., 2018b]: whereby

\textsuperscript{1}http://image-net.org/challenges/LSVRC/2017
In practice, the labeled data may be collected from multiple sources with different distributions [Sun et al., 2015; Bhatt et al., 2016]. In such cases, the aforementioned SDA methods could be trivially applied by combining the sources into a single source: an approach we refer to as source-combined DA. However, source-combined DA oftentimes results in a poorer performance than simply using one of the sources and discarding the others. As illustrated in Figure 2, the accuracy on the best single source digit recognition adaptation using DANN [Ganin et al., 2016] is 71.3%, while the source-combined accuracy drops to 70.8%. For segmentation adaptation using CyCADA [Hoffman et al., 2018b], the mIoU of source-combined DA (37.3%) is also lower than that of SDA from GTA (38.7%). Because the domain shift not only exists between each source and target, but also exists among different sources, the source-combined data from different sources may interfere with each other during the learning process [Riemer et al., 2019]. Therefore, multi-source domain adaptation (MDA) is needed in order to leverage all of the available data.

The early MDA methods mainly focus on shallow models [Sun et al., 2015], either learning a latent feature space for different domains [Sun et al., 2011; Duan et al., 2012] or combining pre-learned source classifiers [Schweikert et al., 2009]. Recently, the emphasis on MDA has shifted to deep learning architectures. In this paper, we systematically survey recent progress on deep learning based MDA, summarize and compare similarities and differences in the approaches, and discuss potential future research directions.

2 Problem Definition

In the typical MDA setting, there are multiple source domains $S_1, S_2, \cdots, S_M$ ($M$ is the number of sources) and one target domain $T$. Suppose the observed data and corresponding labels\(^2\) in the $i^{th}$ source $S_i$ are drawn from distribution $p_i(x, y)$ are $X_i = \{x_i^j\}_{j=1}^{N_i}$ and $Y_i = \{y_i^j\}_{j=1}^{N_i}$, respectively, where $N_i$ is the number of source samples. Let $X_T = \{x_T^j\}_{j=1}^{N_T}$ and $Y_T = \{y_T^j\}_{j=1}^{N_T}$ denote the target data and corresponding labels drawn from the target distribution $P_T(x, y)$, where $N_T$ is the number of target samples.

Suppose the number of labeled target samples is $N_{TL}$, the MDA problem can be classified into different categories:

- **unsupervised MDA**, when $N_{TL} = 0$;
- **fully supervised MDA**, when $N_{TL} = N_T$;
- **semi-supervised MDA**, otherwise.

Suppose $x_i^j \in R^{d_i}$ and $x_T^j \in R^{d_T}$ are an observation in source $S_i$ and target $T$, we can classify MDA into:

- **homogeneous MDA**, when $d_1 = \cdots = d_M = d_T$;
- **heterogeneous MDA**, otherwise.

Suppose $C_i$ and $C_T$ are the label set for source $S_i$ and target $T$, we can define different MDA strategies:

- **closed set MDA**, when $C_1 = \cdots = C_M = C_T$;
- **open set MDA**, otherwise.

\(^2\)The label could be any type, such as object classes, bounding boxes, semantic segmentation, etc.
open set MDA, for at least one $C_i$, $C_i \cap C_T \subset C_T$;

* partial MDA, for at least one $C_i$, $C_T \subset C_i$;

* universal MDA, when no prior knowledge of the label sets is available;

where $\cap$ and $\subset$ indicate the intersection set and proper subset between two sets.

Suppose the number of labeled source samples is $N_{iL}$ for source $S_i$, the MDA problem can be classified into:

* strongly supervised MDA, when $N_{iL} = N_i$ for $i = 1 \ldots M$;

* weakly supervised MDA, otherwise.

When adapting to multiple target domains simultaneously, the task becomes multi-target MDA. When the target data is unavailable during training [Yue et al., 2019], the task is often called multi-source domain generalization or zero-shot MDA.

## 3 Datasets

The datasets for evaluating MDA models usually contain multiple domains with different styles, such as synthetic vs. real, artistic vs. sketchy, which impose large domain shift among different domains. Here we summarize the commonly employed datasets in both computer vision (CV) and natural language processing (NLP) areas, as shown in Table 1.

### Digit recognition

Digits-five includes 5 digit image datasets sampled from different domains, including handwritten MNIST (mt) [LeCun et al., 1998], combined MNIST-M (mm) [Ganin and Lempitsky, 2015] from MNIST and randomly extracted color patches, street image SVHN (sv) [Netzer et al., 2011], Synthetic Digits (sy) [Ganin and Lempitsky, 2015] generated from Windows fonts by various conditions, and handwritten USPS (up) [Hull, 1994]. Usually, 25,000 images are sampled for training and 9,000 for testing in mt, mm, sv, and sy. The entire 9,298 images in up are selected.

### Object classification

Office-31 [Saenko et al., 2010] contains 4,110 images in 31 categories collected from office environments in 3 domains: Amazon (A) with 2,817 images downloaded from amazon.com, Webcam (W) and DSLR (D) with 795 and 498 images taken by web camera and digital SLR camera with different photographic settings.

Office-Caltech [Gong et al., 2013] consists of the 10 overlapping categories shared by Office-31 [Saenko et al., 2010] and Caltech-256 (C) [Griffin et al., 2007]. Totally there are 2,533 images.

Office-Home [Venkateswara et al., 2017] consists of about 15,500 images from 65 categories of everyday objects in office and home settings. There are 4 different domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr) and Real-World images (Rw).

ImageCLEF, originated from ImageCLEF 2014 domain adaptation challenge, consists of 12 object categories shared by ImageNet ILSVRC 2012 (I), Pascal VOC 2012 (P), and C. Totally there are 600 images for each domain with 50 for each category.

PACS [Li et al., 2017] contains 9,991 images of 7 object categories extracted from 4 different domains: Photo (P), Art paintings (A), Cartoon (C) and Sketch (S).

DomainNet [Peng et al., 2019], the largest DA dataset to date for object classification, contains about 600K images from 6 domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. There are totally 345 object categories.

### Sentiment classification of images

Sentilime [Lin et al., 2020] is a DA dataset with 4 domains for binary sentiment classification of images: social Flickr and Instagram (FI) [You et al., 2016], artistic ArtPhoto (AP) [Machajdik and Hanbury, 2010], social Twitter I (TI) [You et al., 2015], and social Twitter II (TII) [Borth et al., 2013]. There are 23,308, 806, 1,269, and 603 images in these 4 domains, respectively.

### Vehicle counting

WebCamT [Zhang et al., 2017] is a vehicle counting dataset from large-scale city camera videos with low resolution, low frame rate, and high occlusion. Totally there are 60,000 frames with vehicle bounding box and count annotations. For MDA, 8 cameras located in different intersections are selected, each with more than 2,000 labeled images. We can view each camera as a domain.

### Scene segmentation

Sim2RealSeg contains 2 synthetic datasets (GTA, SYNTHIA) and 2 real datasets (Cityscapes, BDDS) for segmentation. Cityscapes (CS) [Cordts et al., 2016] contains vehicle-centric urban street images collected from a moving vehicle in 50 cities from Germany and neighboring countries. There are 5,000 images with pixel-wise annotations into 19 classes. BDDS [Yu et al., 2018]

---

| Area          | Task                     | Dataset                  | Reference                         | #D | #S | Labels                          | Short description                                                                 |
|--------------|--------------------------|--------------------------|-----------------------------------|----|----|---------------------------------|---------------------------------------------------------------------------------|
| CV           | object classification    | Office-31 (O)            | Saenko et al.                     | 3  | 4,110| 31 classes                     | images from amazon and taken by different cameras                               |
| CV           | object classification    | Office-Caltech (OC)      | Gong et al.                       | 4  | 2,533| 10 classes                     | overlapping categories from Office-31 and C                                      |
| CV           | object classification    | Office-Home (OH)         | Venkateswara et al.               | 4  | 15,500| 65 classes                     | artistic, clipart, product, and real objects                                   |
| CV           | object classification    | ImageCLEF (IC)           | Challenge et al.                  | 3  | 1,800| 12 classes                     | shared categories from 3 datasets                                              |
| CV           | object classification    | PACS (P)                 | Li et al.                         | 4  | 9,991| 7 classes                      | photographic, artistic, cartoon, and sketchy objects                           |
| CV           | object classification    | DomainNet (DN)           | Peng et al.                       | 6  | 600,000| 345 classes                   | clipart, infographic, artistic, quickdrawn, real, and sketchy objects           |
| NLP          | sentiment classification | Sentilime (SI)           | Machajdik and Hanbury             | 4  | 25,986| 2 classes                      | artistic and social images on visual sentiment                                  |
| NLP          | sentiment classification | WebCamT (WT)            | Zhang et al.                      | 8  | 10,000| vehicle counts                 | each camera used as one domain                                                  |
| NLP          | semantic segmentation    | Sim2RealSeg (S2R)        | Corin et al., Yu et al., Richter et al., Ros et al. | 4  | 49,366| 16 classes              | simulation-to-real adaptation for pixel-wise predictions                     |
| NLP          | part-of-speech tagging   | AmazonReviews (AR)       | Chen et al.                       | 4  | ¥12,000| 2 classes                     | reviews on four kinds of products                                               |
| NLP          | part-of-speech tagging   | MediaReviews (MR)        | Liu et al.                        | 5  | 6897  | 2 classes                      | reviews on products and movies                                                  |
| NLP          | part-of-speech tagging   | SANCL (S)                | Petrov and McDonald               | 3  | 5,250 | tags                           | part-of-speech tagging in 5 web domains                                         |

Table 1: Released and freely available datasets for MDA, where ‘#D’ and ‘#S’ represent the number of domains and the total number of samples usually used for MDA, respectively.
contains 10,000 real-world dash cam video frames with a compatible label space with Cityscapes. GTA [Richter et al., 2016] is a vehicle-egocentric image dataset collected in the high-fidelity rendered computer game GTA-V. It contains 24,966 images (video frames) with 19 classes as Cityscapes. SYNTHIA [Ros et al., 2016] is a large synthetic dataset. To pair with Cityscapes, a subset, named SYNTHIA-RANDCITYSCAPES, is designed with 9,400 images which are automatically annotated with 16 compatible Cityscapes classes, one void class, and some unnamed classes. The common 16 classes are used for MDA.

**Sentiment classification of natural languages.** Amazon Reviews [Chen et al., 2012] is a dataset of reviews on four kinds of products: Books (B), DVDs (D), Electronics (E), and Kitchen appliances (K). Reviews are encoded as 5,000 dimensional feature vectors of unigrams and bigrams and are labeled with binary sentiment. Each source has 2,000 labeled examples, and the target test set has 3,000 to 6,000 examples.

Media Reviews [Liu et al., 2017] contains 16 domains of product reviews and movie reviews for binary sentiment classification. 5 domains with 6,897 labeled samples are usually employed for MDA, including Apparel, Baby, Books, Camera taken from Amazon and MR from Rotten Tomato.

**Part-of-speech tagging.** The SANCL dataset [Petrov and McDonald, 2012] contains part-of-speech tagging annotations in 5 web domains: Emails (E), Weblogs (W), Answers (A), Newsgroups (N), and Reviews (R). 750 sentences from each source are used for training.

Unless otherwise specified, each domain is selected as the target and the rest domains are considered as the sources. For WebCamT, 2 domains are randomly selected as the target. For Sim2RealSeg, MDA is often performed using the simulation-to-real setting [Zhao et al., 2019a], i.e. from synthetic GTA, SYNTHIA to real Cityscapes, BDDS. For SANCL, N, R, and A are used as target domains, while E and W are used as target domains [Guo et al., 2018].

### 4 Deep Multi-source Domain Adaptation

Existing methods on deep MDA primarily focus on the unsupervised, homogeneous, closed set, strongly supervised, one target, and target data available settings. That is, there is one target domain, the target data is unlabeled but available during the training process, the source data is fully labeled, the source and target data are observed in the same data space, and the label sets of all sources and the target are the same. In this paper, we focus on MDA methods under these settings.

There are some theoretical analysis to support existing MDA algorithms. Most theories are based on the seminal theoretical model [Blitzer et al., 2008; Ben-David et al., 2010]. Mansour et al. [2009] assumed that the target distribution can be approximated by a mixture of the $M$ source distributions. Therefore, weighted combination of source classifiers has been widely employed for MDA. Moreover, tighter cross domain generalization bound and more accurate measurements on domain discrepancy can provide intuitions to derive effective MDA algorithms. Hoffman et al. [2018a] derived a novel bound using DC-programming and calculated more accurate combination weights. Zhao et al. [2018a] extended the generalization bound of seminal theoretical model to multiple sources under both classification and regression settings. Besides the domain discrepancy between the target and each source [Hoffman et al., 2018a; Zhao et al., 2018a], Li et al. [2018] also considered the relationship between pairwise sources and derived a tighter bound on weighted multi-source discrepancy. Based on this bound, more relevant source domains can be picked out.

Typically, some task models (e.g. classifiers) are learned based on the labeled source data with corresponding task loss, such as cross-entropy loss for classification. Meanwhile, specific alignments among the source and target domains are conducted to bridge the domain shift so that the learned task models can be better transferred to the target domain. Based on the different alignment strategies, we can classify MDA
Latent space transformation tries to align the latent space (e.g., features) of different domains based on optimizing the discrepancy loss or adversarial loss. Intermediate domain generation explicitly generates an intermediate adapted domain for each source that is indistinguishable from the target domain. The task models are then trained on the adapted domain. Figure 3 summarizes the common overall framework of existing MDA methods.

### 4.1 Latent Space Transformation

The two common methods for aligning the latent spaces of different domains are discrepancy-based methods and adversarial methods. We discuss these two methods below, and Table 2 summarizes key examples of each method.

**Discrepancy-based methods** explicitly measure the discrepancy of the latent spaces (typically features) from different domains by optimizing some specific discrepancy losses, such as maximum mean discrepancy (MMD) [Guo et al., 2018; Zhu et al., 2019], Rényi-divergence [Hoffman et al., 2018a], \(L^2\) distance [Rakshit et al., 2019], and moment distance [Peng et al., 2019]. Guo et al. [2020] claimed that different discrepancies or distances can only provide specific estimates of domain similarities and that each distance has its own characteristics.

**Adversarial methods** try to align the features by making them indistinguishable to a discriminator. Some representative optimized objectives include GAN loss [Xu et al., 2018], \(\mathcal{H}\)-divergence [Zhao et al., 2018a], Wasserstein distance [Li et al., 2018; Wang et al., 2019; Zhao et al., 2020], Cosine distance, MMD, Fisher linear discriminant, and Correlation alignment. Minimizing the discrepancy to align the features among the source and target domains does not introduce any new parameters that must be learned.

### Table 2: Comparison of different latent space transformation methods for MDA

| Reference               | Feature extractor | Feature alignment method | Feature alignment loss | Feature alignment domains | Classifier alignment | #C | Classifier task | Task backbone | Dataset | Result |
|-------------------------|-------------------|--------------------------|------------------------|--------------------------|----------------------|----|----------------|--------------|---------|--------|
| [Mancini et al., 2018]  | shared discrepancy| MMD                      | target and each source | --                       | M                    | 1  | PoS metric     | AlexNet      | OC, FC  | 83.6, 91.8, 85.3 |
| [Guo et al., 2018]      | shared discrepancy| Rényi-divergence         | target and each source | CT loss                  | --                   | 1  | uniform        | AlexNet      | O       | 84.9, 91.1 |
| [Zhu et al., 2019]      | shared discrepancy| MMD                      | target and each source | \(L^1\) loss             | M                    | 1  | --             | ResNet-50    | O, OC, IC| 90.2, 89.4, 74.1 |
| [Rakshit et al., 2019]  | shared discrepancy| \(L^2\) distance         | pairwise all domains   | CT loss                  | --                   | 1  | --             | ResNet-50    | O, OC, IC| 88.3, 97.5, 91.2 |
| [Peng et al., 2019]     | shared discrepancy| moment distance          | pairwise all domains   | \(L^1\) loss             | M                    | 1  | relative error | LNet-5       | D       | 87.7   |
| [Guo et al., 2020]      | shared discrepancy| mixture-distance         | target and each source | CT loss                  | --                   | 1  | --             | BiLSTM       | MR      | 79.4   |
| [Li et al., 2018]       | shared classifier  | Wasserstein              | pairwise all domains   | CT loss                  | --                   | 1  | perplexity score | AlexNet      | D, O, IC| 74.2, 83.5, 80.5 |
| [Zhu et al., 2018a]     | shared classifier  | Wasserstein              | target and each source | --                      | --                   | 1  | --             | AlexNet      | D       | 79.9   |
| [Zhu et al., 2020]      | unshared classifier| Wasserstein              | target and each source | --                      | M                    | -- | Wasserstein    | LNet-5       | D       | 88.1   |

Target prediction. After aligning the features of target
and source domains in the latent space, the classifiers trained based on the labeled source samples can be used to predict the labels of a target sample. Since there are multiple sources, it is possible that they will yield different target predictions. One way to reconcile these different predictions is to uniformly average the predictions from different source classifiers [Zhu et al., 2019]. However, different sources may have different relationships with the target, e.g., one source might better align with the target, so a non-uniform, weighted averaging of the predictions leads to better results. Weighting strategies, known as a source selection process, include uniform weight [Zhu et al., 2019], perplexity score based on adversarial loss [Xu et al., 2018], point-to-set (PoS) metric using Mahalanobis distance [Guo et al., 2018], relative error based on source-only accuracy [Peng et al., 2019], and Wasserstein distance based weights [Zhao et al., 2020].

Besides the source importance, Zhao et al. [2020] also considered the sample importance, i.e., different samples from the same source may still have different similarities from the target samples. The source samples that are closer to the target are distilled (based on a manually selected Wasserstein distance threshold) to fine-tune the source classifiers. Automatically and adaptively selecting the most relevant training samples for each source remains an open research problem.

### 4.2 Intermediate Domain Generation

Feature-level alignment only aligns high-level information, which is insufficient for fine-grained predictions, such as pixel-wise semantic segmentation [Zhao et al., 2019a]. Generating an intermediate adapted domain with pixel-level alignment, typically via GANs [Goodfellow et al., 2014], can help address this problem.

**Domain generator.** Since the original GAN is highly under-constrained, some improved versions are employed, such as Coupled GAN (CoGAN) in [Russo et al., 2019] and CycleGAN in MADAN [Zhao et al., 2019a]. Instead of directly taking the original source data as input to the generator [Russo et al., 2019; Zhao et al., 2019a], Lin et al. [2020] used a variational autoencoder to map all source and target domains to a latent space and then generated an adapted domain from the latent space. Russo et al. [2019] then tried to align the target and each adapted domain, while Lin et al. [2020] aligned the target and combined adapted domain from the latent space. Zhao et al. [2019a] proposed to aggregate different adapted domains using a sub-domain aggregation discriminator and cross-domain cycle discriminator, where the pixel-level alignment is then conducted between the aggregated and target domains. Zhao et al. [2019a] and Lin et al. [2020] showed that the semantics might change in the intermediate representation, and that enforcing a semantic consistency before and after generation can help preserve the labels.

**Feature alignment and target prediction.** Feature-level alignment is often jointly considered with pixel-level alignment. Both alignments are usually achieved by minimizing the GAN loss with a discriminator. One classifier is trained on each adapted domain [Russo et al., 2019] and the multiple predictions for a given target sample are averaged. Only one classifier is trained on the aggregated domain [Zhao et al., 2020] or on the combined adapted domain [Lin et al., 2020] which is obtained by a unique generator from the latent space for all source domains. The comparison of these methods are summarized in Table 3.

### 5 Conclusion and Future Directions

In this paper, we provided a survey of recent MDA developments in the deep learning era. We motivated MDA, defined different MDA strategies, and summarized the datasets that are commonly used for performing MDA evaluation. Our survey focused on a typical MDA setting, i.e., unsupervised, homogeneous, closed set, and one target MDA. We classified these methods into different categories, and compared the representative ones technically and experimentally. We conclude with several open research directions:

**Specific MDA strategy implementation.** As introduced in Section 2, there are many types of MDA strategies, and implementing an MDA strategy according to the specific problem requirement would likely yield better results than a one-size-fits-all MDA approach. Further investigation is needed to determine which MDA strategies work the best for which types of problems. Also, real-world applications may have a small amount of labeled target data; determining how to include this data and what fraction of this data is needed for a certain performance remains an open question.

**Multi-modal MDA.** The labeled source data may be of different modalities, such as LiDAR, radar, and image. Further research is needed to find techniques for fusing different data modalities in MDA. A further extension of this idea is to have varied modalities in different sources as well as partially labeled, multi-modal sources.

**Incremental and online MDA.** Designing incremental and online MDA algorithms remains largely unexplored and may provide great benefit for real-world scenarios, such as updating deployed MDA models when new source or target data becomes available.

| Reference          | Domain generator                  | Pixel alignment domains | Feature alignment loss | Feature alignment domains | #C | Classifier weight | Task backbone | Dataset        | Task | Result |
|--------------------|-----------------------------------|-------------------------|------------------------|---------------------------|----|------------------|--------------|---------------|------|--------|
| [Russo et al., 2019] | CycleGAN shared                    | target and each source  | GAN loss               | target and each source    | M  | uniform          | DeepLabV2     | S2R-CS        | seg  | 42.8   |
| [Zhao et al., 2019a] | CycleGAN shared                    | target and aggregated source | GAN loss              | target and each source    | 1  | —                | FCN8s        | S2R-BDDS      | seg  | 41.4   |
| [Lin et al., 2020]  | VAE+CycleGAN unshared              | target and combined source | —                    | —                          | 1  | —                | ResNet-18     | SI            | cls  | 68.1   |

Table 3: Comparison of different intermediate domain generation methods for MDA, where ‘#C’, ‘seg’, and ‘cls’ are short for the number of classifiers during reference (M is the number of source domains), segmentation, and classification, respectively. ‘Result’ is the average performance of all target domains measured by accuracy for classification and mean intersection-over-union (mIoU) for segmentation.
