Learning Transferable Features from Multi-source Domains with Moment Matching Network

Xingchao Peng
Boston University
xpeng@bu.edu

Qinxun Bai
Hikvision Research America
qinxun.bai@gmail.com

Kate Saenko
Boston University
saenko@bu.edu

Bo Wang
Stanford University
bowang87@cs.stanford.edu

Abstract

Transfer leaning is critical to generalize models learned from one domain to new tasks or situations. Conventional unsupervised transfer learning assumes that training data is sampled from a single domain, whereas a more practical setting where training data are collected from multiple sources is usually neglected. To address this problem, we propose a new approach to transfer knowledge learned from multiple labeled source domains to an unlabeled target domain by dynamically aligning moments of their feature distributions. The key contribution of the proposed method lies in two folds: firstly, we solve the moment matching problem across diverse source domains and train deep neural networks in an end-to-end manner; secondly, we provide a sound theoretical analysis of moment-related error bounds for multi-source domain adaptation. Extensive experimental results further confirm that the proposed method outperforms existing state-of-the-art methods by a large margin.

1 Introduction

Modern deep models have seen a significant performance degradation when trained and tested on different domains, a phenomenon known as *domain shift* or *domain bias* (Quionero-Candela et al., 2009). Recently, transfer learning algorithms have been proposed to learn more generic representations, which allow the models to adapt to novel tasks or domains. The conventional unsupervised transfer learning models assume that there is only a single source domain. However, when we are designing a practical *domain adaptation* (DA) system, it is more likely that we have multiple source domains due to the availability of massive data from the web and social media.

In this paper, we propose a novel approach called Moment Matching Network (MMN), which learns from multiple source domains and transfers the learned knowledge to the target domain. As showed in Figure 1, our model contains three component: feature extractor, moment matching component and classifiers. The feature extractor maps source domains and the target domain to a common latent feature space. The moment matching component aligns the latent feature distributions by matching their moments. When trained in an end-to-end manner, our model is able to alleviate the domain shift between multi-source and target domains.

Compared to state-of-the-art multi-source domain adaptation methods (Xu et al., 2018; Duan, Xu, and Chang, 2012), which only align the multiple source domains to the target domain, our model bridges the source domains to the target domain and also aligns the source domains with each other. We empirically show that matching the source domains is essential for multi-source domain adaptation.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

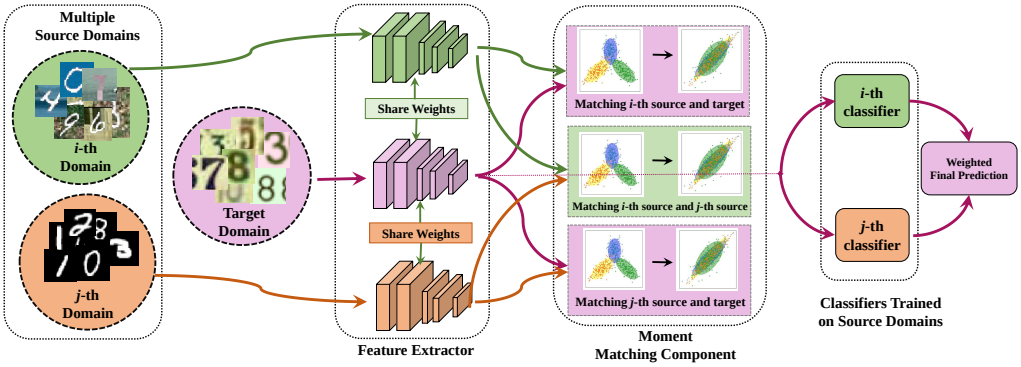


Figure 1: The framework of Moment Matching Network. Our model takes multi-source annotated training data as input and transfers the learned knowledge to classify the unlabeled target samples.

Moreover, inspired by (Saito et al., 2018), we extend our model to further propose MMN-β, which has two hypotheses per source domain. In summary, our contributions are: (1) We propose a novel approach to tackle multi-source domain adaptation by aligning the moments of feature distributions; (2) We derive a theoretical argument relating the target testing error with moments discrepancies between the target domain and multiple source domains.

2 Moment Matching for Multi-source Domain Adaptation

To save space, we propose the definition of multi-source domain adaptation and error bound in Appendix. Theorem 1 (*See Appendix 1*) shows that the upper bound on the target error of the learned hypothesis depends on the pairwise moment divergence between the target domain and each source domain. This motivates our method for multi-source domain adaptation to align the moments between each pair of domains to achieve a better matching result. Given computational concerns in practice, however, our algorithm only aligns moments up to the second order, and adopts the following L-2 norm moment distance. [1]

Definition 1. *Assume* $\mathbf{X}_1$, $\mathbf{X}_2$,..., $\mathbf{X}_N$, $\mathbf{X}_T$ *are collections of i.i.d. samples from* $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N, \mathcal{D}_T$ *respectively, the second order Moment Distance between* $\mathcal{D}_S$ *and* $\mathcal{D}_T$ *is defined as*

$$MD^2(\mathcal{D}_S, \mathcal{D}_T) = \sum_{k=1}^{2} \Big(\frac{1}{N} \sum_{i=1}^{N} \|\mathbb{E}(\mathbf{X}_i^k) - \mathbb{E}(\mathbf{X}_T^k)\|_2 + \binom{n}{2}^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|\mathbb{E}(\mathbf{X}_i^k) - \mathbb{E}(\mathbf{X}_j^k)\|_2 \Big).$$

MMN Our model MMN is based on deep neural networks. Motivated by previous analysis, we propose to train a feature generator G and a set of classifiers $\mathcal{C} = (C_1, C_2, ..., C_N)$, where G aligns all source domains with the target domain, as well as the source domains with each other.

As showed in Figure 1, our model mainly contains three components: feature extractor, moment matching component and classifiers. The N classifiers in our model are trained on the annotated source domains with cross-entropy loss. The objective function is as follows:

$$\min_{G,\mathcal{C}} \sum_{i=1}^{N} \mathcal{L}_{\mathcal{D}_i} + \lambda \min_{G} MD^2(\mathcal{D}_S, \mathcal{D}_T), \tag{1}$$

where $\mathcal{L}_{\mathcal{D}_i}$ is the softmax cross entropy loss for the classifier C_i on domain $\mathcal{D}_i$, λ is the trade-off parameter between the two losses. In testing phase, testing data from the target domain will be forwarded through the feature generator and N classifiers, the final prediction is the averaging of the N classifiers.

Though MMN is capable of aligning the feature distributions from multiple domains, it assumes that $p(y|x)$ will be aligned simultaneously when $p(x)$ is aligned, which will not hold in practice. To mitigate this limitation, we further propose MMN-β.

[1] This can be regarded as an approximation of the second order cross-moment matrix by its trace.

Standards	Models	mt,up,sv,sy → mm	mm,up,sv,sy → mt	mm,mt,sv,sy → up	mm,mt,up,sy → sv	mm,mt,up,sv → 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
	RevGrad	70.81±0.94	97.90±0.83	93.47±0.79	68.50±0.85	87.37±0.68	83.61±0.82
	DAN	67.87±0.75	97.50± 0.62	93.49±0.85	67.80±0.84	86.93±0.93	82.72± 0.79
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
	RevGrad	71.30±0.56	97.60±0.75	92.33±0.85	63.48±0.79	85.34±0.84	82.01±0.76
	CORAL	62.53±0.69	97.21±0.83	93.45±0.82	64.40±0.72	82.77±0.69	80.07±0.75
	DAN	63.78±0.71	96.31±0.54	94.24±0.87	62.45±0.72	85.43±0.77	80.44±0.72
	DCTN	70.53±1.24	96.23±0.82	92.81±0.27	77.61±0.41	86.77±0.78	84.79±0.72
	MMN (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
	MMN-β (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 1: **Digits Classification Results. mt**, **up**, **sv**, **sy**, **mm** are abbreviations for *MNIST*, *USPS*, *SVHN*, *Synthetic Digits*, *MNIST-M*, respectively. Our model MMN-β outperforms other baselines by a large margin.

Standards	Models	A,C,D →W	A,C,W →D	A,D,W →C	C,D,W →A	Avg
Source Combine	Source only	99.0	98.3	87.8	86.1	92.8
	DAN	99.3	98.2	89.7	**94.8**	95.5
Multi-Source	Source only	99.1	98.2	85.4	88.7	92.9
	DAN	**99.5**	99.1	89.2	91.6	94.8
	MMN (ours)	99.4	**99.2**	91.5	94.1	96.1
	MMN-β (ours)	**99.5**	**99.2**	**92.2**	94.5	**96.4**

Table 2: **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.

MMN-β In order to align $p(y|x)$, we follow the training paradigm proposed by Saito et al. (2018). In particular, we leverage two classifiers per domain to form N pairs of classifiers $\mathcal{C}' = ((C_1, C_1'), (C_2, C_2'), ..., (C_N, C_N'))$. Following Saito et al. (2018), we define the discrepancy of two classifiers as the L1-distance between the outputs of the two classifiers. The objective is:

$$\min_{\mathcal{C}'} \sum_{i=1}^{N} \mathcal{L}_{\mathcal{D}_i} - \sum_{i}^{N} |P_{C_i}(D_T) - P_{C_i'}(D_T)|, \tag{2}$$

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).** In the third step, we fix $\mathcal{C}'$ and train G to minimize the discrepancy of each classifier pair on the target domain. The objective function is as follows:

$$\min_{G} \sum_{i}^{N} |P_{C_i}(D_T) - P_{C_i'}(D_T)| \tag{3}$$

The three training steps are performed periodically. We refer our reader to Saito et al. (2018) for the insight behind this training paradigm.

3 Experiments

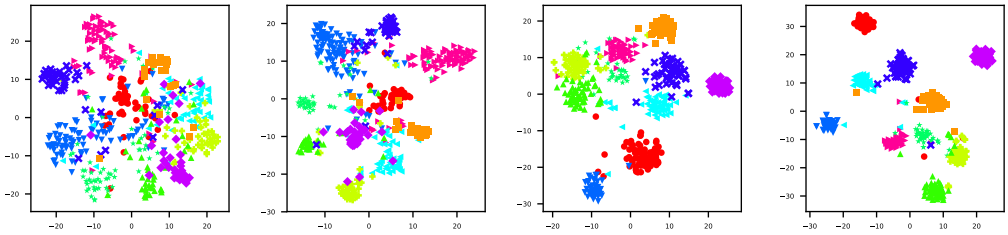
We perform an extensive evaluation on the following tasks: digit classification (*MNIST, SVHN, USPS, MNIST-M, Sythetic Digits*), image recognition and *Office-Caltech10*) and sentimental analysis (Amazon Reviews on *Bools, Electronics, Kitchen Appliances, DVD*).

3.1 Experiments on Digit Recognition

We assess five digit datasets sampled from five different sources, namely *MNIST* (LeCun et al., 1998), *Synthetic Digits* (Ganin et al., 2016), *MNIST-M* (Ganin et al., 2016), *SVHN*, and *USPS*. In our experiments, we take two state-of-the-art discrepancy-based approaches, *i.e.* Deep Adaptation Network (**DAN** Long et al. (2015)) and Correlation Alignment (**CORAL** Sun, Feng, and Saenko (2015)), and two adversarial-based approaches, *i.e.* Reverse Gradient (**RevGrad** Ganin et al. (2016)) and Deep Cocktail Network (**DCTN** Xu et al. (2018)) as our baselines. In the *source combine* setting, all of the source domains are combined into a single domain, and the baseline experiments are conducted in a traditional manner.

Standards	Models	B,D,E →K	B,D,K →E	B,E,K →D	D,E,K →B	Avg
Source combine	Source only	77.9	79.2	77.3	78.2	78.2
	DAN	81.2	82.9	79.3	80.5	80.9
Multi-Source	Source only	75.4	79.3	74.5	72.4	75.4
	DAN	80.4	83.3	77.8	81.1	80.7
	MMN (ours)	**81.7**	86.2	81.2	80.5	82.4
	MMN-β (ours)	81.3	**87.1**	**82.4**	**81.2**	**83.0**

Table 3: **Results on Sentiment Analysis**. B,D,E and K indicate *Books*, *DVD*, *Electronics* and *Kitchen appliances*, respectively. The results show our method outperforms *Deep Adaptation Network* Long et al. (2015) on this task.



(a) DAN Features on SVHN (b) MMN-β Features on SVHN (c) DAN Features on Caltech (d) MMN-β Features on Caltech

Figure 2: Feature visualization: t-SNE plot of DAN features and MMN-β features on SVHN in mm,mt,up,sy→sv setting; t-SNE of DAN features and MMN-β features on Caltech in A,D,W→C setting. We use different markers and different colors to denote different categories. (Best viewed in color.)

The results are shown in Table 1. Our model MMN achieves a **86.13%** average accuracy and MMN-β boosts the performance to **87.65%**, outperforming other baselines by a large margin. For fair comparison, all the experiments are based on the same network architecture.

3.2 Experiments on Office-Caltech10

The Office-Caltech10 (Gong et al., 2012) dataset consists of the same 10 object categories from 4 different domains: *Amazon*, *Caltech*, *DSLR*, and *Webcam*. In our experiments, we choose one domain as the target domain and set the rest as the source domains.

As table 2 shows, our model gets a 96.1% average accuracy on this dataset, and MMN-β further boosts the performance to **96.4**%. All the experiments are based on ResNet-101 pre-trained on ImageNet. We also tried AlexNet, but it does not work as good as ResNet-101.

3.3 Experiments on Sentiment Analysis

In this section, we report experimental results on cross-domain sentiment analysis of text. We use Amazon review dataset (Blitzer, Dredze, and Pereira, 2007). The dataset contains four domains: Kitchen appliances, DVD, Books and Electronics. In each domain, there are 1000 positive and 1000 negative reviews from Amazon users. We leverage a fully connected neural network (with 400-128-32-2 units in each layer) as the sentiment classifier. We apply our model to layer 3, *i.e.* , the layer with 32 units.

Multi-domain adaptation on sentiment analysis is under-explored, so we only compare our model to "source combine" baselines and *Deep Adaptation Network* (Long et al., 2015). Table 3 shows the gain of our model from baselines. Our model MMN achieves a **82.4**% average accuracy across four experiment settings and MMN-β further boosts the result to **83.0**%

Feature visualization To demonstrate the transferability of our model, we visualize the DAN (Long et al., 2015) features and MMN-β features with t-SNE embedding in two tasks, *i.e.* mm,mt,up,sy→sv task and A,D,W→C task. The results are shown in Figure 2. We make two important observations: i) comparing Figure 2(a) with Figure 2(b), we find that MMN-β is capable of learning more discriminative features; ii) from Figure 2(c) and Figure 2(d), we find that the clusters of MMN-β features are tighter than those of DAN, which suggests that the features learned by MMN-β attains more desirable discriminative property. These observations implies the superiority of our model over DAN in multi-source domain adaptation.

References

Anthony, M., and Bartlett, P. L. 2009. *Neural network learning: Theoretical foundations*. cambridge university press.

Ben-David, S.; Blitzer, J.; Crammer, K.; Pereira, F.; et al. 2007. Analysis of representations for domain adaptation. *Proc. NIPS*.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine learning* 79(1-2):151–175.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *ACL*.

Crammer, K.; Kearns, M.; Wortman, J.; and W, J. 2008. Learning from multiple sources. *JMLR* 9(Aug):1757–1774.

Duan, L.; Xu, D.; and Chang, S.-F. 2012. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *CVPR*, 1338–1345. IEEE.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17(1):2096–2030.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2066–2073. IEEE.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*.

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

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

Sun, B.; Feng, J.; and Saenko, K. 2015. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*.

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

4 Appendix

A Multiple Source Domain Adaptation

This section gives the motivation of our approach from a theoretical perspective. Following Ben-David et al. (2010), we first introduce a rigorous model of multi-source domain adaptation for binary classification. A domain $\mathcal{D} = (\mu, f)$ is defined by a probability measure (distribution) μ on the input space $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to [0, 1]$ which can be either deterministic or stochastic. A hypothesis is a function $h : \mathcal{X} \to \{0, 1\}$. The probability that a hypothesis h disagrees with the domain labeling function f under the domain distribution μ is defined as

$$\epsilon_{\mathcal{D}}(h) = \epsilon_{\mathcal{D}}(h, f) = \mathbb{E}_{\mu}[|h(\boldsymbol{x}) - f(\boldsymbol{x})|]. \tag{4}$$

For a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$, we refer to the source error and the target error of a hypothesis h as $\epsilon_S(h) = \epsilon_{\mathcal{D}_S}(h)$ and $\epsilon_T(h) = \epsilon_{\mathcal{D}_T}(h)$ respectively. When the expectation in Equation 4 is computed with respect to an empirical distribution, we denote the corresponding empirical error by $\hat{\epsilon}_{\mathcal{D}}(h)$, such as $\hat{\epsilon}_S(h)$ and $\hat{\epsilon}_T(h)$.

Definition 2 (Multi-Source Domain Adaptation)**.** *Given $\mathcal{D}_S = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}$ the collection of labeled source domains and $\mathcal{D}_T$ 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[2] target error on $\mathcal{D}_T$.*

In order to analyze the target error for multi-source domain adpatation algorithms, it is important to measure the difference between the source domains and the target domain. We therefore introduce the following definition of cross-moment divergence.

Definition 3 (cross-moment divergence)**.** *Given a compact domain $\mathcal{X} \subset \mathbb{R}^n$ and two probability measures μ, μ' on $\mathcal{X}$, the k-th order cross-moment divergence between μ and μ' is*

$$d_{CM^k}(\mu, \mu') = \sum_{\mathbf{i} \in \Delta_k} \Big| \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu(x) - \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu'(x) \Big|,$$

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

Ben-David et al. (2010) provide target error bounds for learning from multiple sources, based on a symmetric difference measure between the target and source domains w.r.t. a hypothesis class. Following the same setup, we provide similar bounds based on the pairwise cross-moment divergence between target and source domains.

In the multi-source settings, a learning algorithm is presented with training sets from each of the N source domains and learns a model that performs well on the target domain. In particular, we examine algorithms that minimize convex combinations of source errors, i.e., given a weight vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ with $\sum_{j=1}^{N} \alpha_j = 1$, we define the $\boldsymbol{\alpha}$-weighted source error of hypothesis h as

$$\epsilon_{\boldsymbol{\alpha}}(h) = \sum_{j=1}^{N} \alpha_j \epsilon_j(h),$$

where $\epsilon_j(h)$ is the shorthand of $\epsilon_{\mathcal{D}_j}(h)$. The empirical $\boldsymbol{\alpha}$-weighted source error can be defined analogously and denoted by $\hat{\epsilon}_{\boldsymbol{\alpha}}(h)$.

Theorem 1. *Let $\mathcal{H}$ be a hypothesis space of VC dimension d. For each $\mathcal{D}_j \in \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}$, let S_j be a labeled sample set of size $\beta_j m$ drawn from μ_j and labeled by the groundtruth labeling function f_j. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_{\boldsymbol{\alpha}}(h)$ for a fixed weight vector $\boldsymbol{\alpha}$ and $h_T^* = \min_{h \in \mathcal{H}} \epsilon_T(h)$ is the target error minimizer, then for any $\delta \in (0, 1)$ and any $\epsilon > 0$, there exist N integers $\{n_\epsilon^j\}_{j=1}^N$ and N constants $\{a_{n_\epsilon^j}\}_{j=1}^N$, such that with probability at least $1 - \delta$,*

$$\epsilon_T(\hat{h}) \leq \ \epsilon_T(h_T^*) + \eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta} + \epsilon + \sum_{j=1}^{N} \alpha_j \Big(2\lambda_j + a_{n_\epsilon^j} \sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_j, \mathcal{D}_T) \Big), \tag{5}$$

[2]The testing setup on $\mathcal{D}_T$ varies across different benchmarks, either closed-set or open-set. In our experiments, we include benchmarks of both types.

where $\eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta} = 4\sqrt{(\sum_{j=1}^{N}\frac{\alpha_j^2}{\beta_j})(\frac{2dlog(2(m+1))+log(\frac{4}{\delta})}{m})}$ *and* $\lambda_j = \min_{h\in\mathcal{H}}\{\epsilon_T(h)+\epsilon_j(h)\}$.

Proof. See Subsection B □

Though the target error bound (Equation. 5) does not explicitly depend on divergences between source domains, it is obvious that the last term of the bound is lower bounded by the divergences between source domains. To see this, consider the toy example consisting of two sources $\mathcal{D}_1$, $\mathcal{D}_2$ and a target domain $\mathcal{D}_T$, it is obvious by Definition. 3 that $d_{CM^k}(\mu,\mu')$ is a metric, which satisfies triangle inequality, therefore,

$$d_{CM^k}(\mathcal{D}_1,\mathcal{D}_T) + d_{CM^k}(\mathcal{D}_2,\mathcal{D}_T) \geqslant d_{CM^k}(\mathcal{D}_1,\mathcal{D}_2). \tag{6}$$

That is the reason why the feature generator G in our proposed model aligns not only all source domains with the target domain, but also the source domains with each other.

B Proof of Theorem 1

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

$$\sup_{\boldsymbol{x}\in\mathcal{C}}|f(\boldsymbol{x}) - P_m(\boldsymbol{x})| \to 0, \quad as\ m\to\infty.$$

Note that for $\boldsymbol{x}\in\mathbb{R}^n$, a multivariate polynomial $P_m : \mathbb{R}^n \to \mathbb{R}$ is of the form

$$P_m(\boldsymbol{x}) = \sum_{k=1}^{m}\sum_{\mathbf{i}\in\Delta_k} a_{\mathbf{i}}\prod_{j=1}^{n}(x_j)^{i_j},$$

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_ϵ and a constant a_{n_ϵ}, such that*

$$|\epsilon_S(h,h') - \epsilon_T(h,h')| \leq \frac{1}{2}a_{n_\epsilon}\sum_{k=1}^{n_\epsilon} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T) + \epsilon.$$

Proof.

$$\begin{aligned}
&|\epsilon_S(h,h') - \epsilon_T(h,h')| \leq \sup_{h,h'\in\mathcal{H}}|\epsilon_S(h,h') - \epsilon_T(h,h')| \\
&= \sup_{h,h'\in\mathcal{H}}|\mathbf{P}_{x\sim D_S}[h(x)\neq h'(x)] - \mathbf{P}_{x\sim D_T}[h(x)\neq h'(x)]| \\
&= \sup_{h,h'\in\mathcal{H}}\left|\int_{\mathcal{X}}\mathbf{1}_{h(x)\neq h'(x)}d\mu_S - \int_{\mathcal{X}}\mathbf{1}_{h(x)\neq h'(x)}d\mu_T\right|,
\end{aligned} \tag{7}$$

where $\mathcal{X}$ is a compact subset of $\mathbb{R}^n$. Given that the indicator function is a Lebesgue integrable function (L^1 function) and that the space of continuous functions with compact support, denoted by $\mathcal{C}_c(\mathcal{X})$, is dense in $L^1(\mathcal{X})$, for any $\frac{\epsilon}{2}>0$, there exists $f\in\mathcal{C}_c(\mathcal{X})$, such that,

$$\begin{aligned}
&\sup_{h,h'\in\mathcal{H}}\left|\int_{\mathcal{X}}\mathbf{1}_{h(x)\neq h'(x)}d\mu_S - \int_{\mathcal{X}}\mathbf{1}_{h(x)\neq h'(x)}d\mu_T\right| \\
&\leq \left|\int_{\mathcal{X}} f(x)d\mu_S - \int_{\mathcal{X}} f(x)d\mu_T\right| + \frac{\epsilon}{2}.
\end{aligned} \tag{8}$$

Using Theorem 2, for any $\frac{\epsilon}{2}$, there exists a polynomial $P_{n_\epsilon} = \sum_{k=1}^{n_\epsilon} \sum_{\mathbf{i} \in \Delta_k} \alpha_{\mathbf{i}} \prod_{j=1}^{n} (x_j)^{i_j}$, such that

$$
\begin{aligned}
&\left| \int_{\mathcal{X}} f(x) d\mu_S - \int_{\mathcal{X}} f(x) d\mu_T \right| \\
\leq &\left| \int_{\mathcal{X}} P_{n_\epsilon} d\mu_S - \int_{\mathcal{X}} P_{n_\epsilon} d\mu_T \right| + \frac{\epsilon}{2} \\
\leq &\sum_{k=1}^{n_\epsilon} \Big| \sum_{\mathbf{i} \in \Delta_k} a_{\mathbf{i}} \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu_S \\
&- \sum_{\mathbf{i} \in \Delta_k} a_{\mathbf{i}} \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu_T \Big| + \frac{\epsilon}{2} \\
\leq &\sum_{k=1}^{n_\epsilon} \sum_{\mathbf{i} \in \Delta_k} \Big(|a_{\mathbf{i}}| \Big| \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu_S \\
&- \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu_T \Big| \Big) + \frac{\epsilon}{2} \\
\leq &\sum_{k=1}^{n_\epsilon} \Big(a_{\Delta_k} \sum_{\mathbf{i} \in \Delta_k} \Big| \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu_S \\
&- \sum_{\mathbf{i} \in \Delta_k} \int_{\mathcal{X}} \prod_{j=1}^{n} (x_j)^{i_j} d\mu_T \Big| \Big) + \frac{\epsilon}{2} \\
= &\sum_{k=1}^{n_\epsilon} a_{\Delta_k} d_{CM^k}(\mathcal{D}_S, \mathcal{D}_T) + \frac{\epsilon}{2} \\
\leq &\frac{1}{2} a_{n_\epsilon} \sum_{k=1}^{n_\epsilon} d_{CM^k}(\mathcal{D}_S, \mathcal{D}_T) + \frac{\epsilon}{2}, \qquad (9)
\end{aligned}
$$

where $a_{\Delta_k} = \max_{\mathbf{i} \in \Delta_k} |a_{\mathbf{i}}|$ and $a_{n_\epsilon} = 2 \max_{1 \leq k \leq n_\epsilon} |a_{\Delta_k}|$. Combining Equation 7, 8, 9, we prove the lemma. □

Lemma 4 (Lemma 6, Ben-David et al. 2010)**.** *For each $\mathcal{D}_j \in \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$, let S_j be a labeled sample set of size $\beta_j m$ drawn from μ_j and labeled by the groundtruth labeling function f_j. For any fixed weight vector $\boldsymbol{\alpha}$, let $\hat{\epsilon}_{\boldsymbol{\alpha}}(h)$ be the empirical $\boldsymbol{\alpha}$-weighted error of some fixed hypothesis h on these sample sets, and let $\epsilon_{\boldsymbol{\alpha}}(h)$ be the true $\boldsymbol{\alpha}$-weighted error. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$:*

$$
\mathbf{P}[|\hat{\epsilon}_{\boldsymbol{\alpha}}(h) - \epsilon_{\alpha}(h)| \geq \epsilon] \leq 2 \exp\Big(\frac{-2m\epsilon^2}{\sum_{j=1}^{N} \frac{\alpha_j^2}{\beta_j}} \Big).
$$

Now we are ready to prove Theorem 1.

Theorem 1. *Let $\mathcal{H}$ be a hypothesis space of VC dimension d. For each $\mathcal{D}_j \in \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}$, let S_j be a labeled sample set of size $\beta_j m$ drawn from μ_j and labeled by the groundtruth labeling function f_j. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_{\boldsymbol{\alpha}}(h)$ for a fixed weight vector $\boldsymbol{\alpha}$ and $h_T^* = \min_{h \in \mathcal{H}} \epsilon_T(h)$ is the target error minimizer, then for any $\delta \in (0, 1)$ and any $\epsilon > 0$, there exist N integers $\{n_\epsilon^j\}_{j=1}^N$ and N constants $\{a_{n_\epsilon^j}\}_{j=1}^N$, such that with probability at least $1 - \delta$,*

$$
\epsilon_T(\hat{h}) \leq \ \epsilon_T(h_T^*) + \eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta} + \epsilon + \sum_{j=1}^{N} \alpha_j \Big(2\lambda_j + a_{n_\epsilon^j} \sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_j, \mathcal{D}_T) \Big), \qquad (5)
$$

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

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

$$
\begin{aligned}
&|\epsilon_{\boldsymbol{\alpha}}(h)-\epsilon_T(h)| \\
&\le \Big|\sum_{j=1}^N \alpha_j\epsilon_j(h)-\epsilon_T(h)\Big| \le \sum_{j=1}^N \alpha_j|\epsilon_j(h)-\epsilon_T(h)| \\
&\le \sum_{j=1}^N \alpha_j\Big(|\epsilon_j(h)-\epsilon_j(h,h_j^*)|+|\epsilon_j(h,h_j^*)-\epsilon_T(h,h_j^*)| \\
&\quad + |\epsilon_T(h,h_j^*)-\epsilon_T(h)|\Big) \\
&\le \sum_{j=1}^N \alpha_j(\epsilon_j(h_j^*)+|\epsilon_j(h,h_j^*)-\epsilon_T(h,h_j^*)|+\epsilon_T(h_j^*)) \\
&\le \sum_{j=1}^N \alpha_j\Big(\lambda_j+\frac{1}{2}a_{n_\epsilon}^j\sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T)\Big)+\frac{\epsilon}{2}. \qquad (10)
\end{aligned}
$$

The third inequality follows from the triangle inequality of classification error[3] (Ben-David et al., 2007; Crammer et al., 2008). The last inequality follows from the definition of λ_j and Lemma 3. Now using both Equation 10 and Lemma 4, we have for any $\delta\in(0,1)$ and any $\epsilon>0$, with probability $1-\delta$,

$$
\begin{aligned}
\epsilon_T(\hat{h}) &\le \epsilon_{\boldsymbol{\alpha}}(\hat{h})+\frac{\epsilon}{2} \\
&\quad +\sum_{j=1}^N \alpha_j\Big(\lambda_j+\frac{1}{2}a_{n_\epsilon}^j\sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T)\Big) \\
&\le \hat{\epsilon}_{\boldsymbol{\alpha}}(\hat{h})+\frac{1}{2}\eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta}+\frac{\epsilon}{2} \\
&\quad +\sum_{j=1}^N \alpha_j\Big(\lambda_j+\frac{1}{2}a_{n_\epsilon}^j\sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T)\Big) \\
&\le \hat{\epsilon}_{\boldsymbol{\alpha}}(h_T^*)+\frac{1}{2}\eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta}+\frac{\epsilon}{2} \\
&\quad +\sum_{j=1}^N \alpha_j\Big(\lambda_j+\frac{1}{2}a_{n_\epsilon}^j\sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T)\Big) \\
&\le \epsilon_{\boldsymbol{\alpha}}(h_T^*)+\eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta}+\frac{\epsilon}{2} \\
&\quad +\sum_{j=1}^N \alpha_j\Big(\lambda_j+\frac{1}{2}a_{n_\epsilon}^j\sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T)\Big) \\
&\le \epsilon_T(h_T^*)+\eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta}+\epsilon \\
&\quad +\sum_{j=1}^N \alpha_j\Big(2\lambda_j+a_{n_\epsilon}^j\sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_S,\mathcal{D}_T)\Big).
\end{aligned}
$$

The first and the last inequalities follow from Equation 10, the second and the fourth inequalities follow from Lemma 4 and the standard sample symmetrization trick for proving growth function/VC dimension bound (Anthony and Bartlett, 2009). The third inequality follows from the definition of $\hat{h}$. □

[3]For any labeling function f_1, f_2, f_3, we have $\epsilon(f_1,f_2)\le\epsilon(f_1,f_3)+\epsilon(f_2,f_3)$.

Standards	Models	I,C→P	I,P→C	P,C→I	Avg
Source combine	Source only	68.3	88.0	81.2	79.2
	RevGrad	67.0	90.7	81.8	79.8
	DAN	68.8	88.8	81.3	79.6
Multi-Source	Source only	68.5	89.3	81.3	79.7
	DCTN	68.8	90.0	83.5	80.8
	MMN (ours)	69.5	90.3	84.9	81.6
	MMN -β (ours)	**70.4**	**91.2**	**85.3**	**82.3**

Table 4: **Results on ImageCLEF-DA**. I, C and P represent *ImageNet ILSVRC 2012*, *Caltech 256* and *Pascal VOC 2012*, respectively. Our model MMN -β performs better than the state-of-the-art models.

	B,D,E→K	B,D,K→E	B,E,K→D	D,E,K→B	Avg
w	81.7	86.2	81.2	80.5	82.4
w/o	80.1	85.8	79.8	78.5	80.3

Table 5: Ablation study on sentiment analysis experiment. The performance will drop if the source domains are not aligned. *w* and *w/o* denote "with aligning the source domains" or "without".

C Experiments on ImageCLEF-DA dataset

ImageCLEF-DA dataset was collected for ImageCLEF 2014 domain adaptation challenge. It contains 12 categories shared in three real-world image datasets, *i.e. ImageNet ILSVRC 2012*, *Pascal VOC 2012*, *Caltech-256*. For each category, 50 images were randomly selected from the original image collections. In total, each domain contains 600 images. In our experiments, we use Reverse Gradient (**RevGrad**, Ganin et al. 2014), Deep Adaptation Network (**DAN**, Long et al. 2015) and Deep Cocktail Network (**DCTN**, Xu et al. 2018) as our baselines.

The experimental results on *ImageCLEF-DA* are shown in table 4. Our model MMN achieves **81.6%** average accuracy, outperforming the state-of-the-art method. The two hypotheses version MMN-β has a 0.7 percent gain from MMN . All the experiments are based on AlexNet architecture.

Ablation Study The moment matching component in our model contains two parts: aligning multi-source domains with target domain and aligning source domains with each other. To demonstrate the importance of matching the source domains, we conduct two set of experiments on sentiment analysis, *i.e. w* or *w/o* align the source domains. The results are shown in Table 5. The performance will drop if the source domains are not aligned.