8. Appendix

The appendix is organized as follows: Section A shows the ablation study for source-source alignment. Section B introduces the formal definition of the cross-moment divergence; Section C gives the proof of Theorem 1 and further discussions; Section D provides the details of experiments on "Digit-Five" dataset; Section E shows feature visualization with t-SNE plot; Section F shows how the number of categories will affect the performance of the state-of-the-art models; Section G and Section H introduce the ResNet baselines and Train/Test split of our DomainNet dataset, respectively; Section I and Section J show the image samples and the detailed statistics of our DomainNet dataset; Section K shows a toy experiment to demonstrate the importance of aligning the source domains; Section L shows the time consumption of our method, compared to baseline.

A. 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.

Schema	digit-five	Office-Caltech10	DomainNet
S-S only	81.5 (+4.1)	94.5 (+1.6)	34.4 (+1.5)
S-T only	85.8 (**+8.1**)	96.2 (**+3.3**)	39.7 (**+6.8**)
M^{3}SDA-β	87.7 (+10)	96.4 (+3.5)	42.6 (+9.7)

Table 6. S-S only: only matching source domains with each other; S-T only: only matching source with target; "+": performance gain from baseline.

B. Cross-moment Divergence

Definition 2 (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*

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

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.

C. 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.

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

where $\mathcal{X}$ is a compact subset of $\mathbb{R}^n$. For any fixed h, h', the indicator function $\mathbf{1}_{h(\boldsymbol{x}) \neq h'(\boldsymbol{x})}(\boldsymbol{x})$ 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,

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

Using Theorem 2, for any $\frac{\epsilon}{2}$, there exists a polynomial

[4] with respect to the corresponding norm

$P_{n_\epsilon}(\boldsymbol{x}) = \sum_{k=1}^{n_\epsilon} \sum_{\mathbf{i}\in\Delta_k} \alpha_{\mathbf{i}} \prod_{j=1}^{n} (x_j)^{i_j}$, such that

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

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. □

Note that the constants a_{Δ_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_ϵ} such that $|P_{n_\epsilon}(\boldsymbol{x}) - f(\boldsymbol{x})| < \epsilon$ and $|a_{\Delta_k}| < \epsilon M_k, \forall k = 1, \ldots, n_\epsilon$.

Lemma 4 (Lemma 6, [1]). *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*

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

Lemma 4 is a slight modification of the Hoeffdings inequality for the empirical $\boldsymbol{\alpha}$-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 m be the size of labeled samples from all sources $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}$, S_j be the labeled sample set of size $\beta_j m$ ($\sum_j \beta_j = 1$) 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 $1 - \delta$,*

$$
\begin{aligned}
\epsilon_T(\hat{h}) \leq &\epsilon_T(h_T^*) + \eta_{\boldsymbol{\alpha},\boldsymbol{\beta},m,\delta} + \epsilon \\
&+ \sum_{j=1}^{N} \alpha_j \Big(2\lambda_j + a_{n_\epsilon^j} \sum_{k=1}^{n_\epsilon^j} d_{CM^k}(\mathcal{D}_j, \mathcal{D}_T) \Big), \qquad (6)
\end{aligned}
$$

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

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

The third inequality follows from the triangle inequality of classification error[5] [2, 4]. 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

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

$\epsilon > 0$, with probability $1-\delta$,

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

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 $\hat{\epsilon}_{\boldsymbol{\alpha}}$ and the true expected error $\epsilon_{\boldsymbol{\alpha}}$. The last term (a summation), as shown in Equation 10, characterizes the upper bound of the difference between the $\boldsymbol{\alpha}$-weighted error $\epsilon_{\boldsymbol{\alpha}}$ and the target error ϵ_T. The constants $\{a_{n^j_\epsilon}\}_{j=1}^N$ 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 $d_{CM^k}(\mathcal{D}_j, \mathcal{D}_T)$, 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.

D. Details of Digit Experiments

Network Architecture In our digital experiments (Table 2), our feature extractor is composed of three *conv* layers and two *fc* layers. We present the *conv* layers as (*input*, *output*, *kernel*, *stride*, *padding*) and *fc* layers as (*input*, *output*). The three *conv* layers are: *conv1* (3, 64, 5, 1, 2), *conv2* (64, 64, 5, 1, 2), *conv3* (64, 128, 5, 1, 2). The two *fc* layers are: *fc1* (8192, 3072), *fc2* (3072, 2048). The architecture of the feature generator is: (*conv1*, *bn1*, *relu1*, *pool1*)-(*conv2*, *bn2*, *relu2*, *pool2*)-(*conv3*, *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 *mm,mt,sv,sy→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.

E. Feature visualization

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

F. Effect of Category Number

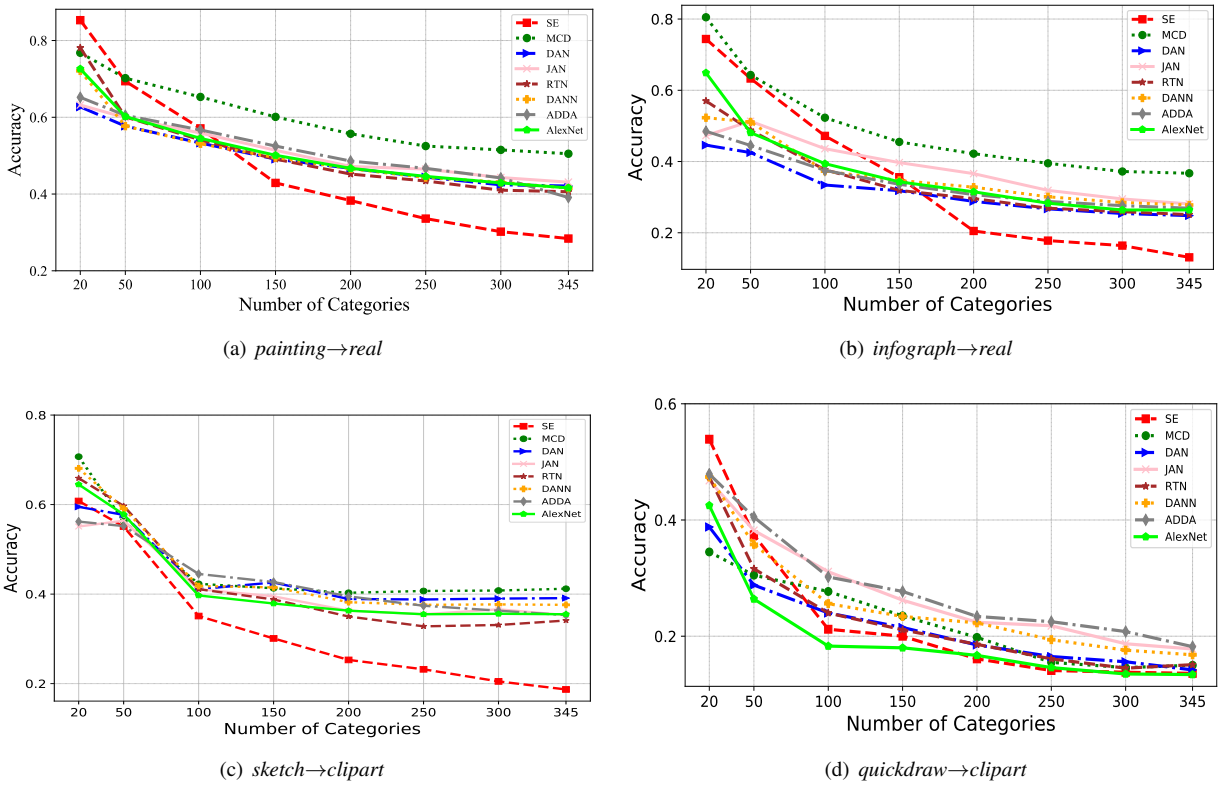
In this section, we show more results to clarify how the number of categories affects the performances of the state-of-the-art models. We choose the following four settings, *i.e.*, *painting→real* (Figure 5(a)), *infograph→real* (Figure 5(b)), *sketch→clipart* (Figure 5(c)), *quickdraw→clipart* (Figure 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 to large scale domain adaptation.

ResNet101	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	19.3	37.5	11.1	52.2	41.0	32.2
inf	30.2	N/A	31.2	3.6	44.0	27.9	27.4
pnt	39.6	18.7	N/A	4.9	54.5	36.3	30.8
qdr	7.0	0.9	1.4	N/A	4.1	8.3	4.3
rel	48.4	22.2	49.4	6.4	N/A	38.8	33.0
skt	46.9	15.4	37.0	10.9	47.0	N/A	31.4
Avg.	34.4	15.3	31.3	7.4	40.4	30.5	**26.5**

ResNet152	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	19.8	37.9	12.2	52.3	44.8	33.4
inf	31.3	N/A	31.1	4.7	45.5	29.6	28.4
pnt	42.0	19.5	N/A	7.4	55.0	37.7	32.3
qdr	12.2	1.8	2.9	N/A	6.3	9.4	6.5
rel	50.5	24.4	49.0	6.2	N/A	39.9	34.0
skt	51.0	18.2	39.7	12.5	47.4	N/A	33.8
Avg.	37.4	16.7	32.1	8.6	41.3	32.3	**28.1**

Table 7. **Single-source *ResNet101* and *ResNet152* [13] baselines on the DomainNet 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*.)



(a) *painting→real* (b) *infograph→real*

(c) *sketch→clipart* (d) *quickdraw→clipart*

Figure 5. **Accuracy *vs.* Number of Catogries**. 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.

G. ResNet baselines

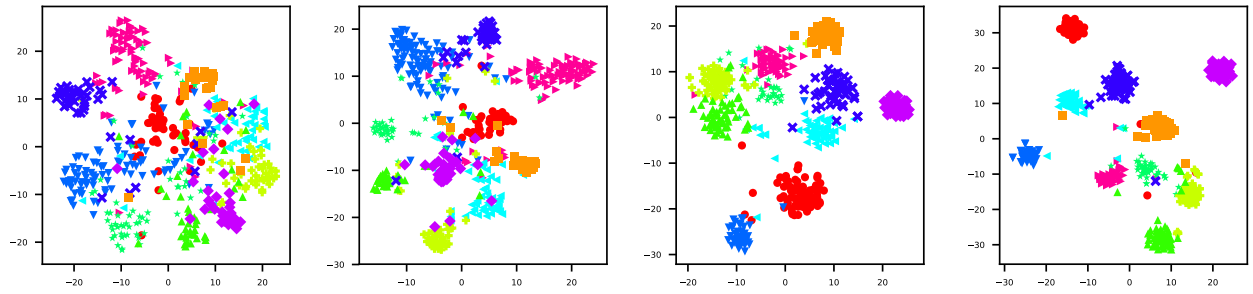
We report ResNet [13] source only baselines in Table 7. The MCD [38] and the SE [7] methods (In Table 3) are based on ResNet101 and ResNet152, respectively. We have observed these two methods both perform worse than their source only baselines, which indicates negative transfer [31] phenomenon occurs 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.

	clp	*inf*	*pnt*	*qdr*	*rel*	*skt*	Total
Train	34,019	37,087	52,867	120,750	122,563	49,115	416,401
Test	14,818	16,114	22,892	51,750	52,764	21,271	179,609
Total	48,837	53,201	75,759	172,500	175,327	70,386	596,010
Per-Class	141	154	219	500	508	204	1,728

Table 8. **Train/Test split.** We split DomainNet with a 70%/30% ratio. The "Per-Class" row shows the average number of images that each category contains.

H. 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.



(a) DAN Features on SVHN (b) M^{3}SDA-β Features on SVHN (c) DAN Features on Caltech (d) M^{3}SDA-β Features on Caltech

Figure 6. Feature visualization: t-SNE plot of DAN features and M^{3}SDA-β features on SVHN in mm,mt,up,sy→sv setting; t-SNE of DAN features and M^{3}SDA-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.)

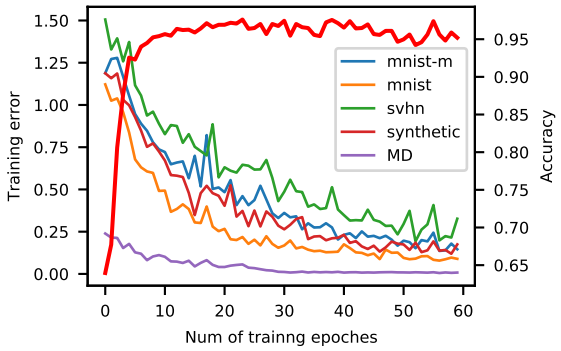
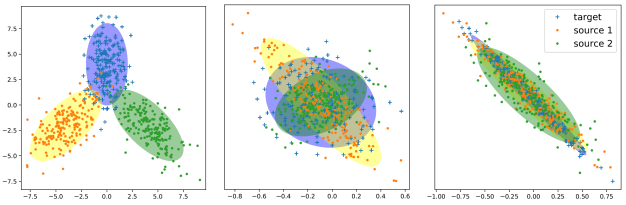


Figure 7. Analysis: training error of each classifier, Moment Discrepancy (MD, defined in Equation 1), and accuracy (red bold line) *w.r.t.* training epochs in *mm,mt,sv,sy→up* setting. The bold red line shows how the accuracy changes *w.r.t.* training epochs.



(a) Original (b) No Within-Source Matching (c) Our model

Figure 8. Yellow and green points are sampled from two source domains. Blue points are sampled from target domain.

I. Image Samples

We sample the images for each domain and show them in Figure 9 (*clipart*), Figure 10 (*infograph*), Figure 11 (*painting*), Figure 12 (*quickdraw*), Figure 13 (*real*), and Figure 14 (*sketch*).

J. Dataset Statistics

Table 10, Table 11, and Table 12 show the detailed statistics of our DomainNet 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*.

Models	Training (ms)	Testing (ms)
ResNet101	200.87	60.84
M^{3}SDA	267.58	61.20

Table 9. Time consumption of our model and the baseline.

K. Toy Experiment

In multi-source domain adaptation, the source domains are not *i.i.d* and as a result, are not automatically aligned with each other. Intuitively, it is not possible to perfectly align the target domain with every source domain, if the source domains are not aligned themselves. More specifically, as explained in the paragraph following Theorem 1, the last term of our target error bound is lower bounded by pairwise divergences between source domains. This motivates our algorithm to also align the moments between each pair of source domains.

Empirically, let's consider a toy experiment setting in which we have two source domains (denoted by yellow and green) and one target domain (denoted by blue), as shown in Figure 8. In the experiments, we utilize a 2-layer fully connected neural network as the backbone. We can observe the source domains and target domain are better aligned when matching source domain distributions using our model is applied.

L. Time Consumption

We show the training and testing time consumption in Figure 9. The experiments are run in PyTorch with a NVIDIA TITAN X GPU on CentOS 7 server. The server has 8 Intel CORE i7 processors. We benchmark all the models with a minibatch size of 16 and an image size of 224 x 224. The CUDA version is 8.0 with CuDNN 5.15.

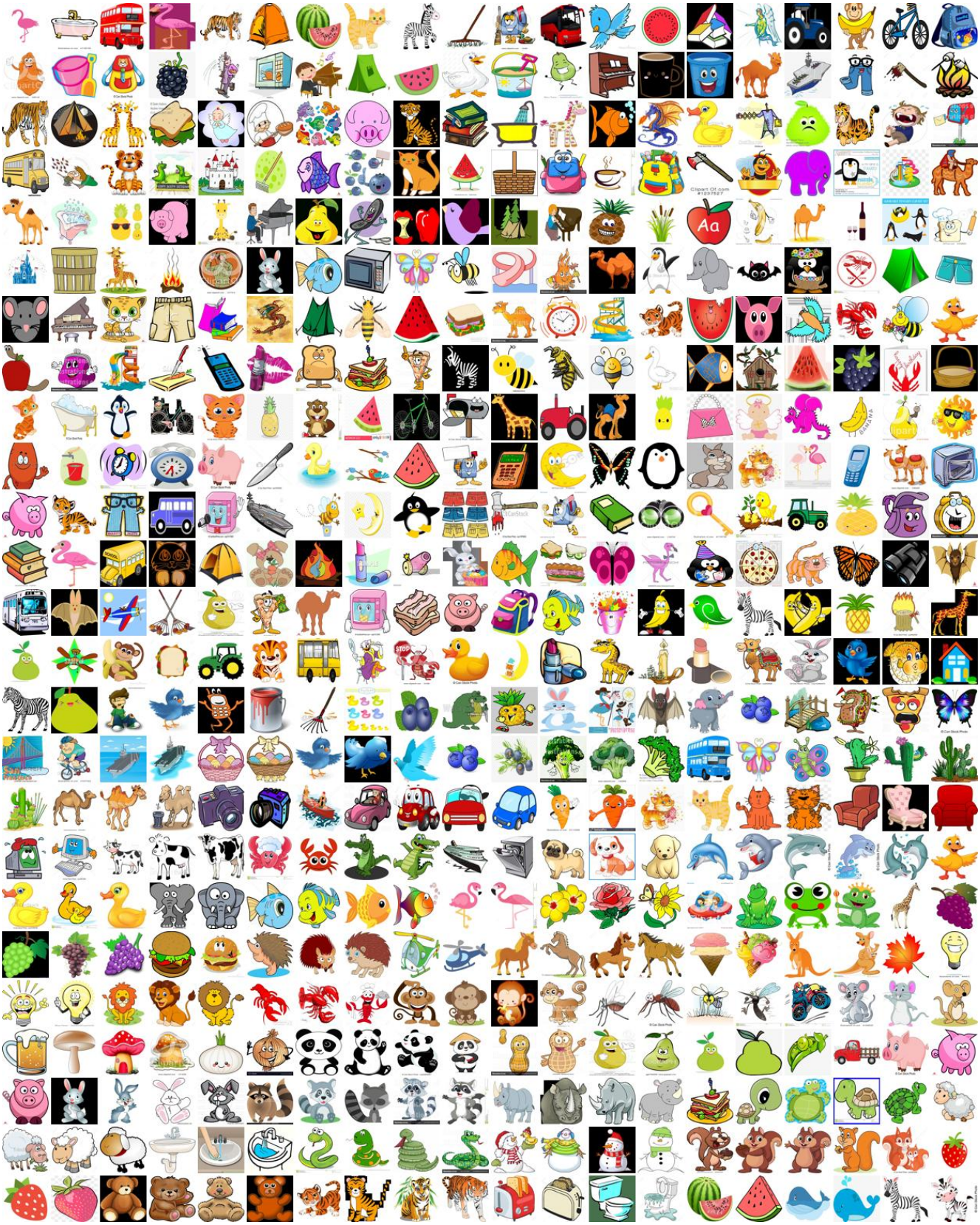


Figure 9. Images sampled from *clipart* domain of the DomainNet dataset.

Figure 10. Images sampled from *infograph* domain of the DomainNet dataset.

Figure 11. Images sampled from *painting* domain of the DomainNet dataset.

Figure 12. Images sampled from *quickdraw* domain of the DomainNet dataset.

Figure 13. Images sampled from *real* domain of the DomainNet dataset.

Figure 14. Images sampled from *sketch* domain of the DomainNet dataset.

Furniture																									
class	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**	**class**	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**	**class**	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**
bathtub	100	135	45	500	517	210	1507	bed	197	180	46	500	724	188	1835	bench	71	47	167	500	662	290	1737
ceiling fan	35	63	38	500	217	25	878	chair	94	148	53	500	320	96	1211	chandelier	223	29	57	500	393	34	1236
couch	232	61	26	500	601	60	1480	door	81	49	347	500	371	361	1709	dresser	41	23	141	500	234	13	952
fence	165	99	49	500	770	140	1723	fireplace	138	98	15	500	700	123	1574	floor lamp	180	100	10	500	246	278	1314
hot tub	144	86	197	500	757	49	1733	ladder	74	96	418	500	442	244	1774	lantern	179	58	218	500	526	40	1521
mailbox	18	45	101	500	595	151	1410	picture frame	88	60	372	500	207	115	1342	pillow	151	170	144	500	656	115	1736
postcard	91	37	88	500	636	49	1401	see saw	299	28	166	500	273	519	1785	sink	133	32	94	500	231	464	1454
sleeping bag	96	14	17	500	406	591	1624	stairs	386	282	27	500	353	525	2073	stove	256	255	16	500	614	269	1910
streetlight	326	113	537	500	463	268	2207	suitcase	377	224	187	500	432	309	2029	swing set	143	35	129	500	556	96	1459
table	297	736	104	500	563	300	2500	teapot	222	209	391	500	631	327	2280	toilet	175	519	31	500	583	118	1926
toothbrush	159	556	11	500	582	235	2043	toothpaste	105	468	31	500	511	198	1813	umbrella	145	511	299	500	362	297	2114
vase	161	319	262	500	632	187	2061	wine glass	220	628	168	500	338	245	2099								
Mammal																									
bat	35	99	306	500	361	160	1461	bear	124	81	379	500	585	178	1847	camel	154	31	289	500	493	130	1597
cat	43	172	344	500	796	130	1985	cow	188	134	156	500	541	17	1536	dog	70	225	721	500	782	311	2609
dolphin	84	165	401	500	581	85	1816	elephant	115	188	425	500	789	266	2283	giraffe	134	172	105	500	594	186	1691
hedgehog	138	48	248	500	727	109	1770	horse	201	216	521	500	645	103	2186	kangaroo	106	60	214	500	613	122	1615
lion	46	64	505	500	516	330	1961	monkey	123	85	405	500	699	166	1978	mouse	74	50	445	500	147	127	1343
panda	87	86	264	500	587	79	1603	pig	93	203	326	500	577	227	1926	rabbit	105	135	269	500	695	94	1798
raccoon	187	24	249	500	676	348	1984	rhinoceros	102	91	220	500	684	183	1780	sheep	114	70	334	500	796	475	2289
squirrel	221	180	779	500	693	389	2762	tiger	315	285	422	500	607	386	2515	whale	343	432	357	500	671	272	2575
zebra	235	306	298	500	683	278	2300																
Tool																									
anvil	122	23	152	500	332	91	1220	axe	48	92	219	500	382	219	1460	bandage	47	322	197	500	399	56	1521
basket	78	219	417	500	444	192	1850	boomerang	92	45	41	500	628	120	1426	bottlecap	118	26	538	500	606	139	1927
broom	171	35	84	500	639	234	1663	bucket	142	56	61	500	335	162	1256	compass	191	36	78	500	272	14	1091
drill	136	44	21	500	573	144	1418	dumbbell	387	86	189	500	581	190	1933	hammer	147	70	46	500	347	71	1181
key	59	68	97	500	229	137	1090	nail	41	256	838	500	674	23	2332	paint can	60	42	172	500	560	34	1368
passport	26	120	34	500	535	97	1312	pliers	38	65	293	500	453	163	1512	rake	119	66	58	500	594	93	1430
rifle	83	149	240	500	520	122	1614	saw	76	34	150	500	118	110	988	screwdriver	205	34	73	500	417	373	1602
shovel	214	17	112	500	450	630	1923	skateboard	263	50	152	500	557	419	1941	stethoscope	343	107	346	500	496	237	2029
stitches	206	285	17	500	207	34	1249	sword	139	124	470	500	591	384	2208	syringe	128	240	10	500	589	222	1689
wheel	133	385	19	500	410	166	1613																
Cloth																									
belt	137	95	17	500	661	125	1535	bowtie	146	95	292	500	533	327	1893	bracelet	293	123	150	500	715	300	2081
camouflage	181	27	72	500	124	69	973	crown	208	17	81	500	170	176	1152	diamond	207	109	17	500	577	117	1527
eyeglasses	201	118	83	500	680	219	1801	flip flops	147	53	206	500	525	120	1551	hat	120	201	400	500	529	77	1827
helmet	163	263	27	500	622	210	1785	jacket	72	82	272	500	457	84	1467	lipstick	101	104	196	500	446	110	1457
necklace	83	115	347	500	697	114	1856	pants	16	173	381	500	398	136	1604	purse	41	119	49	500	544	228	1481
rollerskates	204	49	322	500	493	141	1709	shoe	127	291	260	500	587	645	2410	shorts	140	29	161	500	443	529	1802
sock	167	453	31	500	531	453	2135	sweater	222	92	153	500	579	167	1713	t-shirt	98	320	12	500	367	155	1452
underwear	253	354	12	500	286	132	1537	wristwatch	285	470	18	500	553	224	2050								
Electricity																									
calculator	55	28	12	500	374	69	1038	camera	58	66	156	500	480	109	1369	cell phone	38	170	136	500	520	23	1387
computer	287	97	19	500	362	31	1296	cooler	214	21	13	500	528	90	1366	dishwasher	109	47	107	500	508	40	1311
fan	148	49	16	500	460	66	1239	flashlight	221	62	418	500	461	95	1757	headphones	285	224	181	500	551	188	1929
keyboard	32	95	370	500	503	64	1564	laptop	26	118	161	500	387	319	1511	light bulb	12	185	482	500	262	405	1846
megaphone	73	91	160	500	560	189	1573	microphone	143	70	152	500	562	156	1583	microwave	16	114	10	500	338	170	1148
oven	14	59	11	500	492	176	1252	power outlet	25	76	102	500	620	95	1418	radio	30	101	65	500	398	165	1259
remote control	117	70	111	500	554	47	1399	spreadsheet	187	397	34	500	751	677	2546	stereo	289	334	12	500	211	90	1436
telephone	148	279	78	500	479	255	1739	television	136	546	51	500	400	127	1760	toaster	196	337	107	500	536	267	1943
washing machine	265	519	15	500	466	155	1920																

Table 10. Detailed statistics of the DomainNet dataset.

Building																								
class	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**	**class**	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**	**class**	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**
The Eiffel Tower	114	190	321	500	553	276	1954	The Great Wall	116	80	159	500	530	148	1533	barn	157	150	426	500	313	201	1747
bridge	66	61	471	500	769	335	2202	castle	47	123	225	500	682	56	1633	church	54	20	142	500	668	35	1419
diving board	182	12	127	500	593	71	1485	garden	63	291	213	500	815	98	1980	garden hose	147	48	179	500	534	84	1492
golf club	207	169	650	500	552	695	2773	hospital	95	48	50	500	674	24	1391	house	108	306	105	500	374	144	1537
jail	104	26	54	500	587	94	1365	lighthouse	123	66	411	500	722	384	2206	pond	105	72	603	500	777	95	2152
pool	139	173	90	500	680	103	1685	skyscraper	195	159	179	500	284	466	1783	square	163	211	144	500	98	727	1843
tent	153	234	141	500	590	339	1957	waterslide	159	328	12	500	606	115	1720	windmill	245	372	397	500	635	245	2394
Office																								
alarm clock	93	23	84	500	521	202	1423	backpack	33	341	265	500	439	220	1798	bandage	47	322	197	500	399	56	1521
binoculars	246	55	148	500	402	266	1617	book	167	188	65	500	731	146	1797	calendar	66	54	44	500	176	59	899
candle	151	68	261	500	621	77	1678	clock	69	50	266	500	619	44	1548	coffee cup	357	191	185	500	643	33	1909
crayon	141	41	19	500	512	285	1498	cup	128	52	582	500	406	396	2064	envelope	147	60	291	500	665	183	1846
eraser	138	34	17	500	356	51	1096	map	42	206	423	500	507	193	1871	marker	57	44	103	500	336	240	1280
mug	168	41	500	500	598	186	1993	nail	41	256	838	500	674	23	2332	paintbrush	25	28	365	500	413	75	1406
paper clip	122	25	112	500	549	119	1427	pencil	51	369	183	500	461	26	1590	scissors	205	103	65	500	568	437	1878
Human Body																								
arm	50	129	422	500	235	249	1585	beard	156	93	373	500	728	481	2331	brain	76	283	233	500	724	270	2086
ear	101	58	187	500	348	199	1393	elbow	199	74	97	500	398	155	1423	eye	108	168	292	500	695	489	2252
face	54	110	20	500	696	452	1832	finger	153	71	57	500	625	283	1689	foot	85	111	86	500	558	261	1601
goatee	255	236	129	500	562	219	1901	hand	97	268	262	500	563	264	1954	knee	45	56	130	500	505	273	1509
leg	89	174	178	500	659	145	1745	moustache	222	28	430	500	424	107	1711	mouth	110	103	51	500	457	172	1393
nose	57	226	512	500	362	103	1760	skull	178	29	189	500	329	600	1825	smiley face	113	46	77	500	226	441	1403
toe	85	407	12	500	356	78	1438	tooth	101	473	109	500	257	181	1621								
Road Transportation																								
ambulance	80	20	74	500	623	115	1412	bicycle	71	272	196	500	705	343	2087	bulldozer	111	55	58	500	635	199	1558
bus	101	183	112	500	745	233	1874	car	99	356	45	500	564	145	1709	firetruck	167	39	359	500	562	328	1955
motorbike	42	209	106	500	772	209	1838	pickup truck	46	116	143	500	619	188	1612	police car	104	51	87	500	740	119	1601
roller coaster	143	46	75	500	637	61	1462	school bus	230	142	66	500	478	405	1821	tractor	154	316	183	500	636	263	2052
train	109	373	406	500	681	240	2309	truck	117	678	158	500	673	265	2391	van	207	806	12	500	442	138	2105
Food																								
birthday cake	165	69	119	500	307	233	1393	bread	197	232	315	500	794	276	2314	cake	108	140	172	500	786	77	1783
cookie	97	78	54	500	677	33	1439	donut	139	65	373	500	630	127	1834	hamburger	187	210	147	500	559	185	1788
hot dog	38	138	148	500	644	143	1611	ice cream	160	187	311	500	657	184	1999	lollipop	74	28	252	500	607	106	1567
peanut	84	60	82	500	423	130	1279	pizza	77	157	127	500	600	202	1663	popsicle	288	79	171	500	639	117	1794
sandwich	189	110	139	500	579	132	1649	steak	155	360	50	500	758	238	2061								
Nature																								
beach	105	183	499	500	622	79	1988	cloud	172	142	278	500	324	100	1516	hurricane	92	68	133	500	420	99	1312
lightning	171	68	199	500	560	94	1592	moon	126	195	324	500	568	155	1868	mountain	67	57	319	500	791	195	1929
ocean	54	47	475	500	591	77	1744	rain	71	78	352	500	274	235	1510	rainbow	61	44	84	500	231	46	966
river	134	155	558	500	651	111	2109	snowflake	175	41	405	500	66	460	1647	star	111	204	98	500	61	205	1179
sun	248	352	572	500	161	258	2091	tornado	169	329	373	500	497	211	2079								
Cold Blooded																								
crab	108	50	153	500	717	152	1680	crocodile	164	56	120	500	713	161	1714	fish	130	195	429	500	479	373	2106
frog	163	118	167	500	761	203	1912	lobster	243	47	254	500	649	174	1867	octopus	190	49	331	500	726	149	1945
scorpion	171	53	133	500	447	455	1759	sea turtle	236	190	410	500	621	254	2211	shark	203	279	269	500	183	532	1966
snail	166	18	321	500	465	405	1875	snake	168	57	425	500	501	470	2121	spider	161	154	308	500	593	645	2361
Music																								
cello	93	34	158	500	450	64	1299	clarinet	214	25	89	500	407	33	1268	drums	194	18	205	500	769	214	1900
guitar	103	204	203	500	632	183	1825	harp	258	37	224	500	649	45	1713	piano	20	66	296	500	570	119	1571
saxophone	236	74	358	500	482	310	1960	trombone	227	195	175	500	484	191	1772	trumpet	117	247	122	500	391	188	1565
violin	174	282	203	500	512	203	1874																

Table 11. Detailed statistics of the DomainNet dataset.

Fruit																								
class	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**	**class**	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**	**class**	**clp**	**inf**	**pnt**	**qdr**	**rel**	**skt**	**total**
apple	88	75	445	500	54	181	1343	banana	50	376	359	500	258	204	1747	blackberry	106	214	14	500	568	60	1462
blueberry	171	110	167	500	733	129	1810	grapes	93	171	318	500	734	287	2103	pear	74	115	448	500	438	183	1758
pineapple	83	139	333	500	673	131	1859	strawberry	357	308	530	500	454	198	2347	watermelon	193	401	410	500	671	128	2303
Sport																								
baseball	116	369	122	500	87	48	1242	baseball bat	106	353	145	500	118	196	1418	basketball	61	219	276	500	237	160	1453
flying saucer	233	40	242	500	396	137	1548	hockey puck	188	59	236	500	400	95	1478	hockey stick	155	197	194	500	394	119	1559
snorkel	278	81	179	500	689	397	2124	soccer ball	85	163	268	500	272	377	1665	tennis racquet	187	195	18	500	456	202	1558
yoga	165	447	161	500	371	251	1895																
Tree																								
bush	46	12	67	500	31	626	1282	cactus	119	36	122	500	658	61	1496	flower	253	140	485	500	360	336	2074
grass	148	312	332	500	378	173	1843	house plant	25	292	416	500	484	156	1873	leaf	12	96	504	500	414	378	1904
palm tree	65	277	607	500	333	166	1948	tree	126	511	571	500	536	555	2799								
Bird																								
bird	336	208	222	500	803	306	2375	duck	142	51	419	500	404	276	1792	flamingo	274	39	224	500	542	142	1721
owl	133	114	496	500	757	202	2202	parrot	75	62	336	500	781	266	2020	penguin	121	201	447	500	700	209	2178
swan	469	52	507	500	326	236	2090																
Vegetable																								
asparagus	49	134	408	500	659	209	1959	broccoli	105	229	100	500	679	181	1794	carrot	52	251	265	500	565	34	1667
mushroom	136	298	254	500	788	252	2228	onion	87	62	471	500	599	158	1877	peas	90	120	90	500	680	81	1561
potato	86	61	58	500	608	83	1396	string bean	139	87	70	500	491	68	1355								
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
octagon	29	115	19	500	465	117	1245	squiggle	148	115	674	500	71	442	1950	triangle	183	364	298	500	376	303	2024
zigzag	323	412	110	500	515	144	2004																
Kitchen																								
fork	200	63	84	500	351	176	1374	frying pan	187	68	169	500	399	132	1455	hourglass	100	100	206	500	289	134	1329
knife	32	108	30	500	582	129	1381	lighter	66	27	27	500	587	118	1325	matches	60	19	56	500	333	56	1024
spoon	228	127	158	500	534	406	1953	wine bottle	230	442	59	500	407	274	1912								
Water Transportation																								
aircraft carrier	27	88	133	500	390	63	1201	canoe	68	71	395	500	703	129	1866	cruise ship	208	94	223	500	632	158	1815
sailboat	162	119	322	500	422	361	1886	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	helicopter	145	216	257	500	804	200	2122	hot air balloon	198	48	453	500	732	170	2101
parachute	82	60	140	500	629	233	1644																
Insect																								
ant	81	62	235	500	381	111	1370	bee	202	233	313	500	452	144	1844	butterfly	160	162	387	500	658	249	2116
mosquito	56	232	65	500	562	144	1559																
Others																								
The Mona Lisa	150	112	191	500	289	145	1387	angel	165	17	504	500	31	299	1516	animal migration	235	68	604	500	444	112	1963
campfire	122	53	217	500	489	86	1467	cannon	103	14	54	500	300	93	1064	dragon	105	30	231	500	485	196	1547
feather	268	432	344	500	505	336	2385	fire hydrant	149	59	29	500	579	148	1464	mermaid	207	30	99	500	449	228	1513
snowman	174	123	901	500	114	712	2524	stop sign	169	54	87	500	168	109	1087	teddy-bear	124	407	301	500	528	238	2098
traffic light	211	280	60	500	379	127	1557																

Table 12. Detailed statistics of the DomainNet dataset.