–Supplementary Material–
Large-Scale Cross-Domain Few-Shot Learning

Jiechao Guan[1] , Manli Zhang[1], Zhiwu Lu[2] (✉)

[1] School of Information, Renmin University of China, Beijing, China
[2] Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
luzhiwu@ruc.edu.cn

1 Proofs of Theoretical Results

Proposition 1 *We have* $\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}(\boldsymbol{\omega}'(\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_j^t))^2=\boldsymbol{\omega}'(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})^{'}\boldsymbol{\omega}$, *where* $\mathbf{A}=\begin{pmatrix}\mathbf{1}'_{n_t} & \mathbf{0} & ... & \mathbf{0}\\ \mathbf{0} & \mathbf{1}'_{n_t} & ... & \mathbf{0}\\ ... & ... & ... & ...\\ \mathbf{0} & \mathbf{0} & ... & \mathbf{1}'_{n_t}\end{pmatrix}\in R^{n_s\times(n_t\times n_s)}$, $\mathbf{B}=(\mathbf{I}_{n_t},\mathbf{I}_{n_t},...,\mathbf{I}_{n_t})\in R^{n_t\times(n_t\times n_s)}$, $\mathbf{1}_{n_t}$ *is* n_t*-dimensional vector with all elements* 1*, and* $\mathbf{I}_{n_t}\in R^{n_t\times n_t}$ *is an identity matrix. Therefore, the solution* $\hat{\boldsymbol{\omega}}$ *of Eq. (8) is exactly the smallest eigenvector of* $(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})^{'}$.

Proof. We can easily obtain that:

$$
\begin{aligned}
&\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}(\boldsymbol{\omega}'(\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_j^t))^2\\
&=\boldsymbol{\omega}'\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}(\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_j^t)(\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_j^t)'\boldsymbol{\omega}\\
&=\boldsymbol{\omega}'\sum_{i=1}^{n_s}(\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_1^t,\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_2^t,...,\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_{n_t}^t)(\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_1^t,\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_2^t,...,\hat{\mathbf{z}}_i^s-\hat{\mathbf{z}}_{n_t}^t)'\boldsymbol{\omega}\\
&=\boldsymbol{\omega}'\sum_{i=1}^{n_s}(\hat{\mathbf{z}}_i^s\mathbf{1}'_{n_t}-\hat{\mathbf{Z}}^t)(\hat{\mathbf{z}}_i^s\mathbf{1}'_{n_t}-\hat{\mathbf{Z}}^t)^{'}\boldsymbol{\omega}\\
&=\boldsymbol{\omega}'(\hat{\mathbf{z}}_1^s\mathbf{1}'_{n_t}-\hat{\mathbf{Z}}^t,...,\hat{\mathbf{z}}_{n_s}^s\mathbf{1}'_{n_t}-\hat{\mathbf{Z}}^t)(\hat{\mathbf{z}}_1^s\mathbf{1}'_{n_t}-\hat{\mathbf{Z}}^t,...,\hat{\mathbf{z}}_{n_s}^s\mathbf{1}'_{n_t}-\hat{\mathbf{Z}}^t)^{'}\boldsymbol{\omega}\\
&=\boldsymbol{\omega}'(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})'\boldsymbol{\omega}
\end{aligned}
$$

Under the constraint $\|\boldsymbol{\omega}\|_2=1$, we can find that the optimal $\hat{\boldsymbol{\omega}}$ is exactly the normalized eigenvector of $(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})^{'}$ with the smallest eigenvalue . ■

Proposition 2 $(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})^{'}=\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{A}'\hat{\mathbf{Z}}^{s'}-\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}'\hat{\mathbf{Z}}^{t'}-\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'\hat{\mathbf{Z}}^{s'}+\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{B}'\hat{\mathbf{Z}}^{t'}$. *Since* $\mathbf{A}\mathbf{A}'=n_t\mathbf{I}_{n_s},\mathbf{B}\mathbf{B}'=n_s\mathbf{I}_{n_t},\mathbf{B}\mathbf{A}'=(\mathbf{1}_{n_t},\mathbf{1}_{n_t},...,\mathbf{1}_{n_t})\in R^{n_t\times n_s}$, *and* $\mathbf{A}\mathbf{B}'=(\mathbf{B}\mathbf{A}')'$, *the computation of* $(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})(\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B})^{'}$ *has a linear time cost* $\mathcal{O}(r^2(n_t+n_s))$ $(r\ll n_t+n_s)$.

Proof. It is easy to verify that: $\mathbf{A}\mathbf{A}'=n_t\mathbf{I}_{n_s},\mathbf{B}\mathbf{B}'=n_s\mathbf{I}_{n_t},\mathbf{B}\mathbf{A}'=(\mathbf{1}_{n_t},\mathbf{1}_{n_t},...,\mathbf{1}_{n_t})\in R^{n_t\times n_s},\mathbf{A}\mathbf{B}'\in R^{n_s\times n_t}$. Therefore, the flops costs to calculate $\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{A}'\hat{\mathbf{Z}}^{s'}$ and $\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{B}'\hat{\mathbf{Z}}^{t'}$ are $(2r^2n_s+r^2)$ and $(2r^2n_t+r^2)$, respectively. For $\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'\hat{\mathbf{Z}}^{s'}$, we first calculate $\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'$. Note that every element in matrix $\mathbf{B}\mathbf{A}'$ is 1, ***we just need to calculate the first column vector of*** $\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'$ (i.e. $\hat{\mathbf{Z}}^t\mathbf{1}_{n_t}$) and then assign the result to the rest column vectors of $\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'$. Hence, the flops costs of this process is $(2rn_t)$. After that, computing $(\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}')\hat{\mathbf{Z}}^{s'}$ costs $(2r^2n_s)$ flops. The total flops costs to calculate $\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'\hat{\mathbf{Z}}^{s'}$ is $(2rn_t+2r^2n_s)$. Similarly, the flops costs of $\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}'\hat{\mathbf{Z}}^{t'}$ is $(2rn_s+2r^2n_t)$. Hence, the total flops costs to calculate $(\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{A}'\hat{\mathbf{Z}}^{s'}-\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}'\hat{\mathbf{Z}}^{t'}-\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{A}'\hat{\mathbf{Z}}^{s'}+\hat{\mathbf{Z}}^t\mathbf{B}\mathbf{B}'\hat{\mathbf{Z}}^{t'})$ is $2r(n_t+n_s+2rn_t+2rn_s+3r)\approx 2r(n_t+n_s+2rn_t+2rn_s)=2(r+2r^2)(n_t+n_s)$, which can be viewed as a linear time cost $\mathcal{O}(r^2(n_t+n_s))$ $(r\ll n_t+n_s)$. ■

Remark 1. Let $\mathbf{G}=\hat{\mathbf{Z}}^s\mathbf{A}-\hat{\mathbf{Z}}^t\mathbf{B}$. If we directly calculate $\mathbf{G}$ and then $\mathbf{G}\mathbf{G}'$, the total flops cost is $2rn_tn_s(n_t+n_s+r+1)$ and the computation cost is $\mathcal{O}(rn_tn_s(n_t+n_s))$, which is much higher than that given by **Proposition 2**.

Proposition 3 *According to the eigenvalue decomposition of positive semi-definite matrices, we have* $\mathbf{X}^t\mathbf{X}^{t'}=\mathbf{U}_1\mathbf{\Sigma}_1\mathbf{U}_1'$, $\hat{\mathbf{Z}}^t\hat{\mathbf{Z}}^{t'}=\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{U}_2'$, *and* $\mathbf{Y}^t\mathbf{Y}^{t'}=\mathbf{U}_3\mathbf{\Sigma}_3\mathbf{U}_3'$, *where* $\mathbf{\Sigma}_1=diag(\lambda_1^{(1)},...,\lambda_d^{(1)})$, $\mathbf{\Sigma}_2=diag(\lambda_1^{(2)},...,\lambda_r^{(2)})$, $\mathbf{\Sigma}_3=diag(\lambda_1^{(3)},...,\lambda_k^{(3)})$. *Let* $\mathbf{C}=(\frac{2}{\lambda_i^{(1)}+\eta+\lambda_j^{(2)}})_{d\times r}$ *and* $\mathbf{D}=(\frac{2}{\lambda_l^{(3)}+\eta+\lambda_j^{(2)}})_{k\times r}$. *Both Eq. (9) and Eq. (10) have and only have one solution:*

$$\hat{\mathbf{W}}_X^t=\mathbf{U}_1[(\mathbf{U}_1'\mathbf{X}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2)\odot\mathbf{C}]\mathbf{U}_2' \quad \hat{\mathbf{W}}_Y^t=\mathbf{U}_3[(\mathbf{U}_3'\mathbf{Y}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2)\odot\mathbf{D}]\mathbf{U}_2'$$

where $\odot$ *means Hardamard product of two matrices (i.e. element-wise product).*

Proof. According to the eigenvalue decomposition of positive semi-definite matrices, we have $\mathbf{X}^t\mathbf{X}^{t'}=\mathbf{U}_1\mathbf{\Sigma}_1\mathbf{U}_1'$, $\hat{\mathbf{Z}}^t\hat{\mathbf{Z}}^{t'}=\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{U}_2'$, $\mathbf{Y}^t\mathbf{Y}^{t'}=\mathbf{U}_3\mathbf{\Sigma}_3\mathbf{U}_3'$, Eq. (9) can be rewritten as the following equivalent form:

$$(\mathbf{U}_1\mathbf{\Sigma}_1\mathbf{U}_1'+\eta\mathbf{I})\hat{\mathbf{W}}_{\mathbf{X}}^t+\hat{\mathbf{W}}_{\mathbf{X}}^t(\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{U}_2')=2\mathbf{X}^t\hat{\mathbf{Z}}^{t'}$$

By left multiplying the both sides of the above equation with $\mathbf{U}_1'$ and right multiplying them with $\mathbf{U}_2$, we can obtain the below results:

$$(\mathbf{\Sigma}_1+\eta\mathbf{I})(\mathbf{U}_1'\hat{\mathbf{W}}_X^t\mathbf{U}_2)+(\mathbf{U}_1'\hat{\mathbf{W}}_X^t\mathbf{U}_2)\mathbf{\Sigma}_2=2\mathbf{U}_1'\mathbf{X}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2$$

For positive semi-definiteness of $\mathbf{X}^t\mathbf{X}^{t'}$ and $\hat{\mathbf{Z}}^t\hat{\mathbf{Z}}^{t'}$, we have $\lambda_i^{(1)}\geq 0,\lambda_j^{(2)}\geq 0$, i.e., $(\lambda_i^{(1)}+\eta+\lambda_j^{(2)})>0$. Hence, $(\mathbf{U}_1'\hat{\mathbf{W}}_X^t\mathbf{U}_2)_{ij}=\frac{2(\mathbf{U}_1'\mathbf{X}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2)_{ij}}{\lambda_i^{(1)}+\eta+\lambda_j^{(2)}}$ and $\hat{\mathbf{W}}_X^t=\mathbf{U}_1[(\mathbf{U}_1'\mathbf{X}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2)\odot\mathbf{C}]\mathbf{U}_2'$. Similarly, we can obtain $\hat{\mathbf{W}}_Y^t=\mathbf{U}_3[(\mathbf{U}_3'\mathbf{Y}^t\hat{\mathbf{Z}}^{t'}\mathbf{U}_2)\odot\mathbf{D}]\mathbf{U}_2'$. ■

Proposition 4 *Let* $\mathbf{H} = (1+\gamma+n_s\beta\lambda)\mathbf{I} + \hat{\mathbf{W}}_X^{t'}\hat{\mathbf{W}}_X^t + \gamma\hat{\mathbf{W}}_Y^{t'}\hat{\mathbf{W}}_Y^t + n_s\beta\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}'$. *Since* $\mathbf{B}\mathbf{B}' = n_s\mathbf{I}_{n_t}$ *and* $\mathbf{H}$ *is positive definite, Eq. (11) has and only has one solution:*

$$\hat{\mathbf{Z}}^t = \mathbf{H}^{-1}[2\hat{\mathbf{W}}_X^{t'}\mathbf{X}^t + 2\gamma\hat{\mathbf{W}}_Y^{t'}\mathbf{Y}^t + \beta(\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}' + \lambda\mathbf{I})\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}']$$

Proof. From **Proposition 2**, $\mathbf{B}\mathbf{B}' = n_s\mathbf{I}_{n_t}$. We can rewrite Equation (11) as an equivalent form:

$$\begin{aligned}&[(1+\gamma+n_s\beta\lambda)\mathbf{I} + \hat{\mathbf{W}}_X^{t'}\hat{\mathbf{W}}_X^t + \gamma\hat{\mathbf{W}}_Y^{t'}\hat{\mathbf{W}}_Y^t + n_s\beta\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}']\hat{\mathbf{Z}}^t = \mathbf{H}\hat{\mathbf{Z}}^t \\ =&2\hat{\mathbf{W}}_X^{t'}\mathbf{X}^t + 2\gamma\hat{\mathbf{W}}_Y^{t'}\mathbf{Y}^t + \beta(\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}' + \lambda\mathbf{I})\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}'\end{aligned}$$

It is clear that the symmetric matrix $\mathbf{H}$ is positively definite. Thus Equation (11) has and only has one solution $\hat{\mathbf{Z}}^t = \mathbf{H}^{-1}[2\hat{\mathbf{W}}_X^{t'}\mathbf{X}^t + 2\gamma\hat{\mathbf{W}}_Y^{t'}\mathbf{Y}^t + \beta(\hat{\boldsymbol{\omega}}\hat{\boldsymbol{\omega}}' + \lambda\mathbf{I})\hat{\mathbf{Z}}^s\mathbf{A}\mathbf{B}']$. ■

2 Experiment Details

2.1 Computing Infrastructure

We run Matlab to conduct our TriAE algorithm on a platform with 2 Intel Xeon E5-2609 v3 CPUs and 128G RAM. For other baselines, we use their published codes online to reproduce the results, and most of their implementations are based on PyTorch [1]. We run all deep learning algorithms on a platform with 2 NVIDIA GeForce GTX TITAN X GPUs.

2.2 Sensitivity Analysis of Hyperparameters

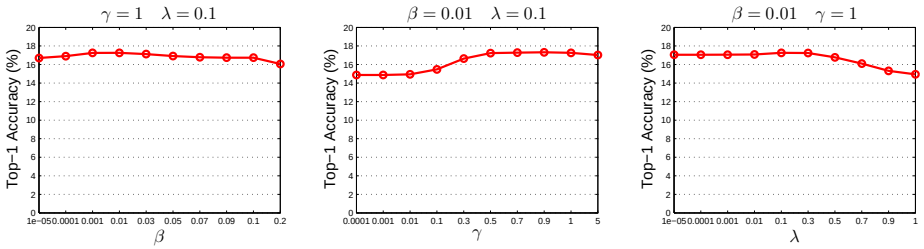

Fig. 1. The influence of three hyperparameters β, γ, λ to the performance of our TriAE model under the large-scale cross-domain 5-shot learning setting.

From Section 4 in main paper, we know that our TriAE model has only three free hyperparameters to tune: β, γ, and λ. β is used to control the weight of domain adaptation loss function and few-shot learning loss function in Eq. (7). γ is the factor to control the importance of the encoder-decoder between the latent subspace and the semantic space. λ is the penalty factor to the difference between

latent representation $\mathbf{Z}^s$ and $\mathbf{Z}^t$. We conduct a group of experiments under the large-scale cross-domain 5-shot learning setting. The influence of these three hyperparameters to TriAE's performance in test stage is presented in Figure 1.

In our experiments, the best hyperparameter setting is: $\beta = 0.01, \gamma = 1, \lambda = 0.1$. For each hyperparameter, we fix the other two ones to evaluate its influence. We have the following observations: (1) As β is approximately equal to zero (i.e. $\beta < 0.0001$), TriAE starts to suffer from performance degradation. This coincides with the results of ablation study in Section 4.3. Since when β is equal to zero, TriAE is simplified as another model (i.e. AE_t1 in Section 4.3) which could not perform as well as TriAE in the test stage. (2) When γ is too small (i.e. $\gamma < 0.01$), TriAE could not perform well. Note that γ is the factor to control the importance of the encoder-decoder between the latent space and the semantic space. The unsatisfactory performance of TriAE with small value γ implies the advantage of introducing the semantic word embedding into our model formulation. (3) When $\lambda \geq 0.3$, the performance of TriAE degrades rapidly. It shows that there does exist a domain shift between the latent representation $\mathbf{Z}^s$ and $\mathbf{Z}^t$, and thus we must select a smaller value of λ.

2.3 Non-overlapped Classes of DomainNet

We choose the Infograph dataset (all infographic images) in DomainNet [2], remove the overlapped ILSVRC2012 1K classes from the original 345 Infograph classes, and leave the non-overlapped 144 classes as the target domain. The corresponding names of 144 classes from DomainNet dataset are shown in Table 1.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (61976220 and 61832017), and Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098).

References

1. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: Advances in Neural Information Processing Systems, Workshop. (2017)
2. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. CoRR **abs/1812.01754** (2018)

Class ID	Class Name	Class ID	Class Name	Class ID	Class Name	Class ID	Class Name
1	fireplace	37	skull	73	teddy-bear	109	finger
2	snowflake	38	skateboard	74	pants	110	bandage
3	scissors	39	tooth	75	streetlight	111	palm tree
4	rollerskates	40	parrot	76	sailboat	112	floor lamp
5	toothpaste	41	watermelon	77	square	113	power outlet
6	bracelet	42	hamburger	78	toothbrush	114	carrot
7	calendar	43	hot air balloon	79	cake	115	peas
8	pond	44	house plant	80	ceiling fan	116	The Eiffel Tower
9	hurricane	45	paint can	81	garden hose	117	crayon
10	grapes	46	cooler	82	The Great Wall of China	118	picture frame
11	campfire	47	suitcase	83	goatee	119	circle
12	fire hydrant	48	shorts	84	windmill	120	calculator
13	moustache	49	stereo	85	bread	121	snowman
14	zigzag	50	bulldozer	86	sea turtle	122	roller coaster
15	motorbike	51	alarm clock	87	paper clip	123	donut
16	postcard	52	stop sign	88	hexagon	124	dresser
17	anvil	53	cell phone	89	steak	125	beard
18	matches	54	spreadsheet	90	squiggle	126	hockey stick
19	blackberry	55	flip flops	91	cloud	127	lightning
20	flying saucer	56	giraffe	92	jail	128	octopus
21	smiley face	57	sink	93	octagon	129	stairs
22	boomerang	58	mermaid	94	animal migration	130	cookie
23	wine glass	59	eyeglasses	95	sweater	131	hospital
24	cruise ship	60	baseball bat	96	stitches	132	ladder
25	onion	61	rainbow	97	elbow	133	moon
26	triangle	62	wristwatch	98	tornado	134	airplane
27	map	63	diving board	99	swing set	135	birthday cake
28	angel	64	raccoon	100	hot tub	136	firetruck
29	chandelier	65	sandwich	101	see saw	137	passport
30	crown	66	headphones	102	light bulb	138	golf club
31	bush	67	camouflage	103	yoga	139	asparagus
32	helicopter	68	The Mona Lisa	104	marker	140	peanut
33	tennis racquet	69	police car	105	megaphone	141	flashlight
34	dolphin	70	blueberry	106	waterslide	142	t-shirt
35	pliers	71	skyscraper	107	axe	143	coffee cup
36	cactus	72	underwear	108	string bean	144	clarinet

Table 1. The set of 144 classes from the DomainNet dataset that has no overlap with the ImageNet2012 1K classes.