Supplementary material for Domain Generalization via Semi-supervised Meta Learning

Hossein Sharifi-Noghabi, Hossein Asghari, Nazanin Mehrasa, Martin Ester
School of Computing Science, Simon Fraser University
Burnaby, British Columbia, Canada
[hsharifi,hasghari,nmehrasa,ester]@sfu.ca

1 Datasets details

PACS benchmark [1] includes four domains: Photo, Sketch, Cartoon, and Art. Each domain has seven common categories: dog, elephant, giraffe. guitar, horse, house, and person. The total number of images is 10046. Photo has 1683 images, Sketch has 3942 images, Cartoon has 2357 images, and Art has 2061 images. We downloaded this dataset and its train/validation/test filelists from `http://www.eecs.qmul.ac.uk/~dl307/project_iccv2017`.

VLCS benchmark [2] aggregates four domains: Caltech-101 [3], PASCAL VOC 2007 [4], LabelMe [5], and Sun09 [6]. The total number of images is 10765. Each domain has five common categories: bird, car, chair, dog, and person. Caltech has 1424 images, PASCAL has 3385 images, Labelme has 2665 images, and Sun has 3291 images. We downloaded this dataset and its train/validation/test filelists from the Github repository of [7].

2 Implementations details

We adopted the same hyper-parameters for the baselines as the original studies [8, 9]. For DGSML, We trained the network for 1000 iterations using the mini-batch stochastic gradient descent (SGD) with a momentum of 0.9. We set the labeled batch size to 128 and set the batch size for the unlabeled samples based on the unlabeled samples rate. For the rate> 0.5. we set the unlabeled batch size to 64, for the rate$= 0.05$ and rate< 0.5, we set it to 32 (16 for rate< 0.5 in PACS). This way, we exploited most of the available samples in each scenario. The meta-train learning rate (α_0) was set to $1e-3$ and the meta-test learning rate (α_1) was set to $1e-4$ for PACS, and $1e-3$ and $1e-6$ for VLCS. The regularization coefficients for the semi-supervised (β_0), the alignment loss (β_1) were set to $1e-3$ and $1e-2$, respectively. For PACS experiments, weight decay was set to $5e-4$. All of the images were preprocessed similar to [7]. We set $|D_{tr}| = 2$ and $|D_{ts}| = 1$ in our experiments.

For the AlexNet experiments, we tuned some of the hyper-parameters of DGSML for small ranges using the validation set. For the unlabeled batch-size, we explored $\{16, 32, 64, 128\}$, for the labeled batch-size we explored $\{64, 128\}$, for the meta-train learning rate we explored $\{1e-4, 1e-3\}$ and for the meta-test learning rate we explored $\{1e-6, 1e-5, 1e-4, 1e-3\}$. For the semi-supervised loss regularization coefficient we explored $\{1e-4, 1e-3, 1e-2\}$. For the global loss regularization coefficient we explored $\{1e-4, 1e-3, 1e-2\}$. We did not tune the weight decay for PACS, number of iterations, optimizer, the momentum, and $|D_{tr}|/|D_{ts}|$ (number of randomly splitted meta-train and meta-test domains). Experiment for each rate of unlabeled samples was repeated five times using five different random seeds including, $\{1, 42, 420, 4200, 42000\}$.

We did not tune the hyper-parameters for DGSML ResNet-18 experiments and adopted the same hyper-parameters as the AlexNet experiments in addition to an exponential learning rate decay with a factor of 0.95 every 10 steps. For MASF, since the implementation for ResNet-18 was not available we implemented from scratch using the Pytorch framework. The proposed hyper-parameters by the authors of this method for AlexNet were not able to reproduce their results, therefore, we tuned the

Preprint.

Table S1: Comparison of DGSML accuracy and fully supervised baselines on VLCS using AlexNet

Source	Target	MASF [9]	DeepAll	DGSML 50%	DGSML 20%
C,L,P	Sun	64.97 ± 0.61	65.09 ± 0.18	$\mathbf{65.27 \pm 0.38}$	$\mathbf{65.73 \pm 0.51}$
L,P,S	Caltech	94.22 ± 0.49	95.21 ± 0.33	$\mathbf{95.93 \pm 0.33}$	$\mathbf{95.49 \pm 0.40}$
C,P,S	Labelme	$\mathbf{61.03 \pm 0.95}$	56.08 ± 0.44	58.43 ± 0.61	58.90 ± 0.66
C,L,S	Pascal	$\mathbf{68.58 \pm 0.40}$	66.98 ± 0.59	67.32 ± 0.43	68.06 ± 0.37
Average		72.02	70.84	71.74	72.05

Note: MASF and DeepAll were trained in a fully supervised way with 0% rate of unlabeled samples.

optimizer and meta learning rates. We used the SGD optimizer with momentum of 0.9, meta-train, meta-test, and metric subnetwork learning rates were set to $1e-3$. We used similar regularization coefficients, and weight decay of the original paper and a similar learning rate decay and iterations as the DGSML method.

For SSL-ProtoNet, we trained the model for 50 epochs and each epoch had 100 iterations (episodes) with Adam optimizer and learning rate of $1e-3$ which decays by a factor of 0.5 every 20 steps. The network contains four convolutions blocks, where each consists of a 64 lters of 3×3 convolution with padding of one, followed by a batch-normalization layer, a ReLU non-linearity, and a 2×2 max-pooling layer, respectively. We used all of the classes (7 for PACS and 5 for VLCS) to train the model. We set number of support and query examples per class to 5 and we followed the same setting for evaluation. This means that we utilized a 5-shot 7-way approach to train the model on PACS and a 5-shot 5-way approach to train it on VLCS. The unlabeled samples were added to each episode based on the rate of the unlabeled samples.

3 Full labeled results

We compared the accuracy of DGSML, trained on 50% and 20% of unlabeled samples (50% and 80% of labeled samples) of the VLCS benchmark, to the fully supervised baselines (DeepAll and MASF), trained on the all of the labeled samples available (0% rate of unlabeled samples), and presented the results in Table S1. It is important to note that MASF and DeepAll cannot incorporate unlabeled samples. Moreover, their performance in the fully labeled scenario should be considered as an upper bound for DGSML because unlike our method, they have access to all of the labels. In two out of four domains (Caltech and Sun), DGSML was able to outperformed the baselines which means that the DGSML semi-supervised domain generalization approach works even better than when the baselines had access to all of the labeled samples. MASF also achieved a better performance than its fully supervised version in one domain (Pascal-see Table 1 in the main paper). Surprisingly, the fully supervised DeepAll was not able to outperform DGSML or MASF. This emphasizes on the role of strong alignment and unlabeled samples in generalization.

4 Ablation study

Table S2 presents the results for the ablation study to investigate the impact of the semi-supervised loss and the alignment loss. We repeated the ablation experiments five times for a given rate of unlabeled samples using similar random seeds that we used for AlexNet and ResNet experiments on the PACS and VLCS datasets. First, we removed the semi-supervised loss which means DGSML had only the task-specific loss and the alignment loss (denoted as $w/o\ l_{sl}$ in table 3) and then we removed the alignment loss which means DGSML had only classifier and the semi-supervised loss (denoted as $w/o\ l_{alignment}$). We reported the average accuracy over all rates of unlabeled samples for each dataset. Our results indicate that DGSML when employing all of its components obtained the best performance, although the accuracy gains achieved by adding certain components were fairly small. On average, all variants of DGSML achieved that performance with a low standard error, which shows their robustness.

5 ResNet-18 detailed results

Table S3 reports the performance of DGSML and the baselines using a ResNet-18 which is a deeper backbone compared to the AlexNet and 4 convolutional layers of SSL-ProtoNet. We observed significant improvements in the performance of DGSML, MASF, and SSL-ProtoNet. Particularly for SSL-ProtoNet the improvement is extremely high which aligns with previous reports regarding methods of few-shot learning [10]. To our surprise, the performance of DeepAll decreases compared to the AlexNet experiments. This can be due to co-adaptation in the outer most layers of the ResNet-18 or transferability of the extracted features in standard fully supervised networks [11].

References

[1] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[2] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[3] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. 2007.

[5] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77 (1-3):157–173, 2008.

[6] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 129–136. IEEE, 2010.

[7] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[8] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. *arXiv preprint arXiv:1911.07661*, 2019.

[9] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6447–6458, 2019.

[10] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[11] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[12] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Table S2: Average accuracy on PACS and VLCS for rates of unlabeled samples in ablation study of DGSML using AlexNet

Dataset	w/o l_{sl}	w/o $l_{alignment}$	DGSML
PACS	64.08 ± 0.91	64.22 ± 0.88	64.44 ± 0.79
VLCS	70.40 ± 0.55	70.76 ± 0.53	70.86 ± 0.58

Table S3: Accuracy on PACS with different rates of unlabeled samples using ResNet-18

Rate	Source	Target	MASF [9]	SSL-ProtoNet [12]	DeepAll	DGSML
20%	A,C,S	Photo	94.12 ± 0.11	86.41 ± 0.22	90.32 ± 0.22	$\mathbf{95.14 \pm 0.14}$
	C,S,P	Art	68.35 ± 0.39	55.13 ± 0.71	59.71 ± 0.15	$\mathbf{72.57 \pm 0.10}$
	A,C,P	Sketch	49.83 ± 0.39	$\mathbf{54.97 \pm 2.58}$	46.68 ± 1.16	53.09 ± 0.34
	A,S,P	Cartoon	70.02 ± 0.53	64.21 ± 0.87	58.09 ± 0.89	$\mathbf{73.62 \pm 0.13}$
50%	A,C,S	Photo	94.40 ± 0.18	85.42 ± 0.40	89.60 ± 0.34	$\mathbf{95.08 \pm 0.07}$
	C,S,P	Art	68.34 ± 0.26	53.03 ± 1.18	59.42 ± 0.65	$\mathbf{72.54 \pm 0.24}$
	A,C,P	Sketch	49.75 ± 0.32	$\mathbf{55.45 \pm 1.66}$	45.62 ± 0.84	52.41 ± 0.53
	A,S,P	Cartoon	69.46 ± 0.39	61.95 ± 0.23	57.90 ± 1.19	$\mathbf{73.52 \pm 0.10}$
80%	A,C,S	Photo	93.99 ± 0.16	82.05 ± 1.30	88.56 ± 0.11	$\mathbf{95.01 \pm 0.10}$
	C,S,P	Art	67.91 ± 0.28	50.70 ± 0.84	57.71 ± 1.04	$\mathbf{72.03 \pm 0.24}$
	A,C,P	Sketch	49.32 ± 0.70	$\mathbf{53.60 \pm 1.06}$	45.28 ± 1.40	52.37 ± 0.91
	A,S,P	Cartoon	69.77 ± 0.45	61.86 ± 0.70	55.67 ± 1.47	$\mathbf{73.30 \pm 0.32}$
95%	A,C,S	Photo	92.87 ± 0.39	74.66 ± 1.17	81.39 ± 0.68	$\mathbf{94.12 \pm 0.21}$
	C,S,P	Art	66.47 ± 0.67	44.92 ± 1.41	52.54 ± 1.88	$\mathbf{70.55 \pm 0.20}$
	A,C,P	Sketch	47.45 ± 0.46	$\mathbf{54.60 \pm 1.35}$	44.95 ± 3.08	49.69 ± 0.51
	A,S,P	Cartoon	69.31 ± 0.99	57.16 ± 2.06	50.13 ± 2.22	$\mathbf{72.42 \pm 0.83}$
Average			70.09	62.22	61.35	72.98