# Supplementary Material

## 1. Implementation Details

### 1.1. Network Architecture

ResNet10 [7] is the backbone of our ProD. The input image size is $3 \times 224 \times 224$. The output of ResNet10 before the classification head serves as a backbone feature, whose size is $7 \times 7 \times 512$ and is reshaped into $49 \times 512$ (49 feature tokens) later to be concatenated with the prompts. The architecture of the transformer layers following the backbone is the same as the transformer unit in [4] (standard ViT transformer unit) with one trasnformer layer and one MLP layer. The size of the DS prompt and DG prompts are set as $5$, and their embedding dimensions are set as $512$, the same as the backbone feature. Concatenated with the prompts, the shape of the input for the transformer layers is $59 \times 512$. The hidden embedding size for the transformer is 512, and the number of transformer heads is $8$. After the transformer layer, the size of the output vector is $59 \times 512$. Finally, the output of DS and DG prompts (both with size $5 \times 512$) are averaged respectively to classify the image.

### 1.2. Multi-domain Training Scheme

We use miniImageNet [11] as our base source dataset since it contains images of immense variety. Four fine-grained cross-domain datasets, including CUB [1], Cars [8], Plantae [12] and Places [13], are selected as other source datasets or target dataset. When one of the four fine-grained cross-domain datasets serves as a target domain dataset for inference, the other three serve as source domain datasets for training.

In each training batch, half of the samples are from the base dataset miniImageNet and another half from a cross-domain dataset. Thus, different mini-batch contains images from a different domain, and the three-domain from the four datasets (CUB, Cars, Plantae, Places) serve as the source domain of the mini-batch in turn. To be compatible with the DS prompt, ”2k” samples are selected in each training batch, where ”k” is the number of categories in a mini-batch. For each category, two samples are selected. For instance, when the batch size is 64, we have $k = 32$. When inference, only one domain other than the three domains selected as the source is sampled for support and query sets.

### 1.3. Evaluation Protocol

Following [2, 5], we evaluate our model by sampling 600 independent 5-way few-shot classifications on the four cross-domain datasets. In each sampled test, $K$ images from $5$ novel categories are selected as a support set whose labels are available for training or fine-tuning. 15 images from the 5 novel categories are selected as a query set whose labels are not available and cannot be used to train or fine-tune the model. Following the standard setting [2,5], we let $K = 1, 5$. The last linear classification head is re-trained from scratch based on the support set, and the rest parameters are frozen when inference. Then the model performance is evaluated on the query set with all the parameters frozen. For each independent test, the linear classification head is re-trained. Statistic information of query images is only used for batch normalization [2, 5]. The model is evaluated 600 times in each experiment, and the average accuracy with a $95\%$ confidence interval, beginning with $\pm$, is reported as the model performance.

### 1.4. Default Hyper-parameters

The default hyper-parameters setting is shown in Tab.2. The model is trained with a batch size of 64 for 500 epochs. The loss weight parameter $\alpha$ is set as $1$, $\beta$ is set as $1$, and the domain center momentum update rate $\lambda$ is set as $0.9$. Transformer depth is set as 2, and both DS and DG prompt sizes are $5$. The model is optimized with adaptive moment estimation (ADAM), with a learning rate of $10^{-2}$ and momentum of $0.9$. When inference, the linear classifier is optimized with stochastic gradient descent (SGD), with a learning rate of $10^{-2}$ and trained for $100$ epochs.

## 2. Additional Experiments

### 2.1. Effect of the Multi-domain Training Scheme

We evaluate the effect of the multi-domain training scheme described in Sec.1.2. The result is shown in Tab.1, where methods with “$^{-}$” are not trained with the multi-domain training scheme. From the result, we see that the multi-domain training scheme increases most methods’ accuracy. It is worth noticing that for DSL, the performance significantly drops after removing the scheme since the model architecture is designed based on the multi-domain

| Methods | CUB | CARS | Plantae | Places |
|---|---|---|---|---|
| RelationNet | $51.10 \pm 0.62$ | $38.26 \pm 0.58$ | $62.99 \pm 0.62$ | $46.01 \pm 0.57$ |
| RelationNet$^-$ | $51.02 \pm 0.64$ | $37.98 \pm 0.59$ | $62.78 \pm 0.61$ | $46.02 \pm 0.56$ |
| MatchingNet | $57.21 \pm 0.63$ | $36.98 \pm 0.56$ | $62.83 \pm 0.62$ | $43.68 \pm 0.55$ |
| MatchingNet$^-$ | $56.92 \pm 0.61$ | $36.94 \pm 0.54$ | $62.51 \pm 0.64$ | $43.51 \pm 0.57$ |
| RelationNet+LFT | $65.02 \pm 0.55$ | $43.51 \pm 0.51$ | $50.48 \pm 0.46$ | $67.34 \pm 0.52$ |
| MatchingNet+LFT | $61.44 \pm 0.56$ | $43.12 \pm 0.52$ | $48.49 \pm 0.51$ | $65.09 \pm 0.48$ |
| RelationNet+ATA | $59.42 \pm 0.48$ | $42.99 \pm 0.42$ | $45.51 \pm 0.51$ | $67.10 \pm 0.41$ |
| NSAE [10] | $68.17 \pm 0.54$ | $54.77 \pm 0.56$ | $59.51 \pm 0.55$ | $70.93 \pm 0.54$ |
| DSL$^-$ | $63.76 \pm 0.60$ | $51.21 \pm 0.40$ | $53.27 \pm 0.49$ | $66.12 \pm 0.78$ |
| DSL | $73.57 \pm 0.65$ | $58.53 \pm 0.73$ | $62.10 \pm 0.75$ | $74.10 \pm 0.72$ |
| Baseline$^-$ | $70.98 \pm 0.76$ | $50.63 \pm 0.72$ | $58.25 \pm 0.69$ | $67.01 \pm 0.57$ |
| Baseline | $72.32 \pm 0.77$ | $53.17 \pm 0.71$ | $60.05 \pm 0.69$ | $69.13 \pm 0.60$ |
| ProD$^-$ | $78.01 \pm 0.79$ | $57.22 \pm 0.63$ | $63.62 \pm 0.68$ | $72.43 \pm 0.63$ |
| ProD | $\mathbf{79.19 \pm 0.59}$ | $\mathbf{59.49 \pm 0.68}$ | $\mathbf{65.82 \pm 0.65}$ | $\mathbf{75.00 \pm 0.72}$ |

Table 1. Comparison with the state of the arts on 5-way 5-shot task with/ without multi-domain training scheme. "$^-$" means the multi-domain training scheme is removed from the corresponding method.

| parameter | value |
|---|---|
| $\alpha$ | 1 |
| $\beta$ | 1 |
| $\lambda$ | 0.9 |
| transformer head | 8 |
| transformer hidden embedding | 512 |
| transformer depth | 2 |
| learning rate train | $10^{-2}$ |
| learning rate inference | $10^{-2}$ |
| training batch size | 64 |
| training epoch | 500 |
| inference re-train epoch | 100 |

Table 2. Default hyper-parameters setting.

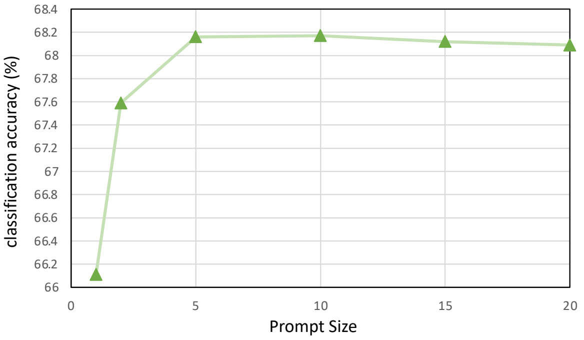

Figure 1. Evaluation of DS prompt sizes for 10-way 5-shot test on CUB.

| Weight ($\alpha$) | CUB | |
|---|---|---|
| | 1-shot | 5-shot |
| 0.01 | $53.12 \pm 0.72$ | $78.75 \pm 0.69$ |
| 0.1 | $53.79 \pm 0.67$ | $79.03 \pm 0.61$ |
| 1 | $53.97 \pm 0.71$ | $79.19 \pm 0.63$ |
| 10 | $52.87 \pm 0.73$ | $78.54 \pm 0.70$ |

Table 3. Evaluation of different weights for neutralizing loss.

training scheme.

In ProD, removing the multi-domain training scheme prevents the DS prompt from learning effective domain-specific knowledge and thus causes the accuracy decline. For instance, on the CUB, the 5-way 5-shot accuracy drops by $-1.18\%$.

### 2.2. DS Prompt Size on 10-way 5-shot Test

We test different DS prompt sizes for 10-way 5-shot test. The result is shown in Fig.1. We draw two following observations:

First, when the DS prompt size increases, the achieved accuracy undergoes a sharp increase and a slight decrease. The reason is twofold: 1) the small Ds prompt cannot include enough sample features to present a novel domain. 2) 5 samples are enough to represent a domain under a 10-way test, indicating that the prompt size does not have to be significantly increased with the category number $C$.

Second, the 10-way test result is lower than the 5-way test since the 10-way task is harder. For each category, we have the same amount of support samples to train the network, but the category number $C$ is increased to 10.

| Methods | ChestX | ISIC | EuroSAT | CropDisease |
|---|---|---|---|---|
| Transductive Ft [6] | 26.79 | 49.68 | 81.76 | 90.64 |
| ConFeSS [3] | 27.09 | 48.85 | 84.65 | 88.88 |
| RDC-FT [9]$^-$ | 25.48 | 49.06 | 84.67 | **93.55** |
| ProD | **28.79** | **50.57** | **85.09** | 90.41 |

Table 4. Comparison with the state of the arts on 5-way 5-shot task on newly proposed datasets.

### 2.3. ProD on New Datasets

As shown in Tab.4, ProD surpasses several newly proposed methods by a clear margin on ChestX, ISIC, and EuroSAT datasets.

### 2.4. Weight of Neutralizing Loss

We test different weight for neutralizing loss: 0.01, 0.1, 1 and 10 in this section. The result is shown in Table 3. The result shows that when the weight is too small, the domain bias within the DG prompt cannot be properly removed. When the weight is too high, the discriminative of the DG prompt declines dramatically. Thus, for all the other experiments, we set the weight $\alpha$ as 1.

## References

[1] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, pages 438–451. Springer, 2010. 1

[2] 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. 1

[3] Debasmit Das, Sungrack Yun, and Fatih Porikli. ConfeSS: A framework for single source cross-domain few-shot learning. In *International Conference on Learning Representations*, 2022. 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 1

[5] Yunhui Guo, Noel Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogério Feris. A broader study of cross-domain few-shot learning. In *ECCV (27)*, pages 124–141, 2020. 1

[6] Yunhui Guo, Noel C. Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning, 2019. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1

[9] Pan Li, Shaogang Gong, Chengjie Wang, and Yanwei Fu. Ranking distance calibration for cross-domain few-shot learning, 2021. 3

[10] Hanwen Liang, Qiong Zhang, Peng Dai, and Juwei Lu. Boosting the generalization capability in cross-domain few-shot learning via noise-enhanced supervised autoencoder. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9404–9414, 2021. 2

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[12] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1

[13] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 1