Domain-Adaptive Few-Shot Learning – Supplemental Material –

An Zhao[1,2*] Mingyu Ding[3*] Zhiwu Lu[1,2†] Tao Xiang[4] Yulei Niu[5] Jiechao Guan[1] Ji-Rong Wen[1]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]The University of Hong Kong [4]University of Surrey, United Kingdom
[5]Nanyang Technological University, Singapore
zhaoan_ruc@163.com mingyuding@hku.hk luzhiwu@ruc.edu.cn

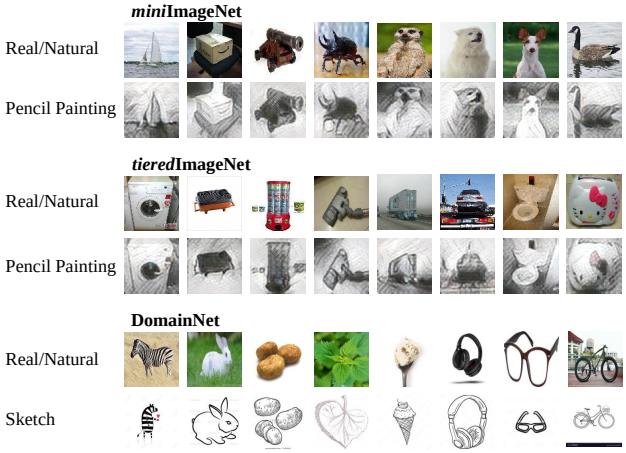


Figure 1. Examples of the source and target data for the three datasets. The domain gap is found to be large for each dataset.

In this document, we provide more supporting materials for our main paper. Firstly, we give the formula derivation of the joint loss $L(\mathbf{W}, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$ in Section 3.4. Secondly, we present examples of the source and target data to show the domain gap in the three datasets. Finally, we give the ablative results for our feature embedding module.

1. Formula Derivation Details

In our main paper, in order to determine the weights of the objectives automatically, we propose an adaptive re-weighting module (see Section 3.4), which adopts an adaptive multi-task loss function based on maximizing the Gaussian likelihood with task-dependent uncertainty.

Formally, our DAPN has four discrete outputs $\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_4$, modeled with multiple softmax likelihoods, respectively. The joint loss $L(\mathbf{W}, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$ is defined as follows:

$$
\begin{aligned}
&L(\mathrm{W}, \sigma_1, \sigma_2, \sigma_3, \sigma_4) \\
&= \mathrm{softmax}(\mathrm{y}_1{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_1) \cdot \mathrm{softmax}(\mathrm{y}_2{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_2) \\
&\quad \cdot \mathrm{softmax}(\mathrm{y}_3{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_3) \cdot \mathrm{softmax}(\mathrm{y}_4{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_4) \\
&= -\log p(\mathrm{y}_1|\mathrm{f}^{\mathrm{W}}(x), \sigma_1) - \log p(\mathrm{y}_2|\mathrm{f}^{\mathrm{W}}(x), \sigma_2) \\
&\quad - \log p(\mathrm{y}_3|\mathrm{f}^{\mathrm{W}}(x), \sigma_3) - \log p(\mathrm{y}_4|\mathrm{f}^{\mathrm{W}}(x), \sigma_4) \\
&= \frac{1}{\sigma_1^2} L_1(\mathrm{W}) + \frac{1}{\sigma_2^2} L_2(\mathrm{W}) + \frac{1}{\sigma_3^2} L_3(\mathrm{W}) + \frac{1}{\sigma_4^2} L_4(\mathrm{W}) \\
&+ \log \frac{\sum_{c'} \exp(\frac{1}{\sigma_1^2} f_{c'}^{\mathrm{W}}(x))}{(\sum_{c'} \exp(f_{c'}^{\mathrm{W}}(x)))^{\frac{1}{\sigma_1^2}}} + \log \frac{\sum_{c'} \exp(\frac{1}{\sigma_2^2} f_{c'}^{\mathrm{W}}(x))}{(\sum_{c'} \exp(f_{c'}^{\mathrm{W}}(x)))^{\frac{1}{\sigma_2^2}}} \\
&+ \log \frac{\sum_{c'} \exp(\frac{1}{\sigma_3^2} f_{c'}^{\mathrm{W}}(x))}{(\sum_{c'} \exp(f_{c'}^{\mathrm{W}}(x)))^{\frac{1}{\sigma_3^2}}} + \log \frac{\sum_{c'} \exp(\frac{1}{\sigma_4^2} f_{c'}^{\mathrm{W}}(x))}{(\sum_{c'} \exp(f_{c'}^{\mathrm{W}}(x)))^{\frac{1}{\sigma_4^2}}} \\
&\approx \frac{1}{\sigma_1^2} L_1(\mathrm{W}) + \frac{1}{\sigma_2^2} L_2(\mathrm{W}) + \frac{1}{\sigma_3^2} L_3(\mathrm{W}) + \frac{1}{\sigma_4^2} L_4(\mathrm{W}) \\
&\quad + \log \sigma_1 + \log \sigma_2 + \log \sigma_3 + \log \sigma_4,
\end{aligned}
\tag{1}
$$

where the classification likelihood is used to to squash a scaled version of the model's output y (i.e. $\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3$, or y_4) with a softmax function:

$$
p(\mathrm{y}|\mathbf{f}^{\mathbf{W}}(x)) = \mathrm{softmax}(\mathbf{f}^{\mathbf{W}}(x)). \tag{2}
$$

More specifically, with a positive scalar σ, the log likelihood for the output y is:

$$
\log p(\mathrm{y} = c|\mathrm{f}^{\mathrm{W}}(x), \sigma) = \frac{1}{\sigma^2} f_c^{\mathrm{W}}(x) - \log \sum_{c'} \exp(\frac{1}{\sigma^2} f_{c'}^{\mathrm{W}}(x)). \tag{3}
$$

2. Illustration of Source and Target Data

As mentioned in our main paper, we select three datasets for problem definition: ***mini*ImageNet** [3], ***tiered*ImageNet** [4], and **DomainNet** [2]. For the first two datasets, the

*Equal Contribution
†Corresponding Author

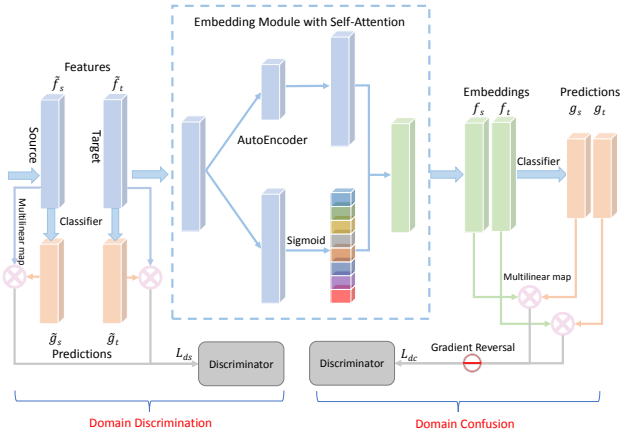


Figure 2. Schematic of the proposed adversarial learning method for domain-adaptive FSL. Both source/target domain confusion and domain discrimination are explicitly included in our model.

Model	5-way 1-shot	5-way 5-shot
Encoder	26.78 ± 0.24	36.42 ± 0.26
Encoder+Decoder	27.07 ± 0.24	36.81 ± 0.28
Encoder+Decoder+Attention	$\mathbf{27.25} \pm 0.25$	$\mathbf{37.45} \pm 0.25$

Table 1. Ablative accuracies (%, top-1) with 95% confidence intervals for our feature embedding module over ***mini*ImageNet**.

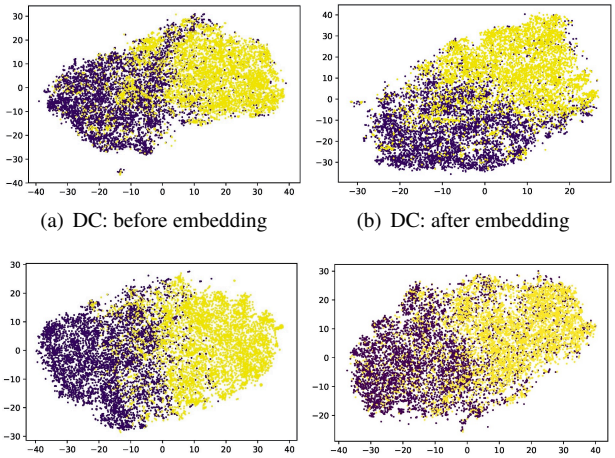
source data are real/natural images, while the target data (i.e., pencil paintings) are obtained by applying the style transfer algorithm [5]. For the third dataset, the real split in DomainNet is used as the source data, while the sketch split is used as the target data. Further, we define our new domain-adaptive few-shot learning (DA-FSL) on these datasets. Figure 1 shows examples of the source and target data for the three datasets. It can be clearly seen that the domain gap between the source and target data is large for each dataset. Therefore, our DA-FSL problem is rather hard to solve since both domain adaptation and FSL are involved.

3. Ablative Results for Feature Embedding

The feature embedding module in our DAPN model consists of three components: encoder, decoder, and attention, as shown in Figure 2. The ablative results for this feature embedding module are shown in Table 1. We can observe that: (1) Adding more components leads to more performance improvements, showing the contribution of each component of our feature embedding module. (2) The feature embedding module used for DA-FSL needs to take a complex form so that both domain confusion and domain discrimination can be enforced simultaneously.

4. Feature Visualization for Our DAA Module

The t-SNE visualization [1] of the feature vectors extracted before/after the embedding module can be seen in Figure 3. It shows that the use of the loss L_{ds} leads to two improvements: (1) The source/target samples are discriminated significantly better before embedding (see Figure 3(c) vs. Figure 3(a)); (2) The source/target samples are enforced to be more confused after embedding (see Figure 3(d) vs. Figure 3(b)). This explains the better performance of our DAA module (w.r.t. the conventional domain confusion).



(a) DC: before embedding (b) DC: after embedding

(c) DC+DS: before embedding (d) DC+DS: after embedding

Figure 3. The t-SNE visualization of the feature vectors of 5,000 randomly-selected images from the source domain (purple dots) and 5,000 images from the target domain (yellow dots) on the DomainNet dataset. *Left*: feature vectors extracted before the embedding module; *Right*: feature vectors extracted after the embedding module. Notations: DC – domain adaptation using the loss L_{dc}; DC+DS – domain adaptation using the losses L_{dc} and L_{ds}.

References

[1] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[2] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. *arXiv preprint arXiv:1812.01754*, 2018.

[3] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[4] 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. In *ICLR*, 2018.

[5] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.