Bi-level Meta-learning for Few-shot Domain Generalization

Xiaorong Qin[1,2], Xinhang Song[1,2], Shuqiang Jiang[1,2]
[1]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS), Institute of Computing Technology, Beijing [2]University of Chinese Academy of Sciences, Beijing
{xiaorong.qin, xinhang.song}@vipl.ict.ac.cn
sqjiang@ict.ac.cn

1. Proof of Lemma1

Lemma 1.1. *Assume that $\mathcal{L}_{D_k}$ is differentiable and $(\mathbf{W}_k^*, \boldsymbol{\phi}_k^*)$ is the unique minimizer of $\mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k, \boldsymbol{\phi}_k) + \frac{\lambda}{2}\|g(\mathbf{W}_k) - g(\mathbf{W}_C)\|_F^2$. Then the gradient components of the meta-loss $F_k(\boldsymbol{\theta}, \mathbf{W}_C)$ with respect to $\boldsymbol{\theta}$ and $\mathbf{W}_C$ are given by $\frac{\partial F_k}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\theta}}$ and $\frac{\partial F_k}{\partial \mathbf{W}_C} = \lambda(\frac{\partial g(\mathbf{W}_C)}{\partial \mathbf{W}_C})^\top[g(\mathbf{W}_C) - g(\mathbf{W}_k^*)]$, which are no Hessian information.*

Proof. First, since $\mathcal{L}_{D_k}$ is differentiable and $(\mathbf{W}_k^*, \boldsymbol{\phi}_k^*) = \arg\min_{\mathbf{W}_k, \boldsymbol{\phi}_k} \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k, \boldsymbol{\phi}_k) + \frac{\lambda}{2}\|g(\mathbf{W}_k) - g(\mathbf{W}_C)\|_F^2$, from the first-order optimality condition we know that, when $\mathbf{W}_k = \mathbf{W}_k^*$ and $\boldsymbol{\phi}_k = \boldsymbol{\phi}_k^*$:

$$
\begin{aligned}
&\frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \mathbf{W}_k^*} + \lambda(\frac{\partial g(\mathbf{W}_k^*)}{\partial \mathbf{W}_k^*})^\top[g(\mathbf{W}_k^*) - g(\mathbf{W}_C)] \\
&= 0, \\
&\frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\phi}_k^*} = 0.
\end{aligned} \tag{1}
$$

Next, we know that:

$$
\begin{aligned}
&F_k(\boldsymbol{\theta}, \mathbf{W}_C) \\
&= \min_{\mathbf{W}_k, \boldsymbol{\phi}_k} \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k, \boldsymbol{\phi}_k) + \frac{\lambda}{2}\|g(\mathbf{W}_k) - g(\mathbf{W}_C)\|_F^2 \\
&= \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*) + \frac{\lambda}{2}\|g(\mathbf{W}_k^*) - g(\mathbf{W}_C)\|_F^2,
\end{aligned} \tag{2}
$$

where $(\mathbf{W}_k^*, \boldsymbol{\phi}_k^*) = \arg\min_{\mathbf{W}_k, \boldsymbol{\phi}_k} \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k, \boldsymbol{\phi}_k) + \frac{\lambda}{2}\|g(\mathbf{W}_k) - g(\mathbf{W}_C)\|_F^2$.

Then, we compute the gradient components of $F_k(\boldsymbol{\theta}, \mathbf{W}_C)$ with respect to $\boldsymbol{\theta}$ and $\mathbf{W}_C$. From the chain rule, we have (3) and (4), where ① and ② hold according to (1).

□

2. Dataset and Implementation details

2.1. More details of Meta-Dataset

Meta-Dataset [3] contains different types of datasets with different categories. Some datasets contain natural images, like ImageNet, Flowers and Birds, but other datasets consist of special types of images. Quick Draw and Omniglot, their images are some black handwritten characters on a white background. Images of textures present perceptual features with different sense of quality, not having a single object in the image. Traffic signs involves a variety of traffic signs. MSCOCO is similar to ImageNet, but lower resolution.

2.2. Implementation details

In this section, we first introduce the implementation details of our baseline model multiple SDLs (single-domain models) and MDL (a multi-domain model) by optimizing (5) and (6), respectively, in our experimental comparison part. We use ResNet-18 as backbone, in line with the other models.

$$
\min_{\boldsymbol{\theta}, \boldsymbol{\phi}_k} \mathcal{L}_{D_k}(\boldsymbol{\theta}, \boldsymbol{\phi}_k),\ k = 1, 2, ..., N. \tag{5}
$$

$$
\min_{\boldsymbol{\theta}, \{\boldsymbol{\phi}_k\}_{k=1}^N} \sum_{k=1}^N \mathcal{L}_{D_k}(\boldsymbol{\theta}, \boldsymbol{\phi}_k). \tag{6}
$$

In our experiment, N equals 8, and the eight datasets are ImageNet, Omniglot, Aircraft, Birds, Textures, Quick Draw, Fungi, VGG Flower from Meta-dataset respectively, also called seen datasets.

We follow the training protocol of precious works [1, 2]. For SDLs, we use SGD with momentum and adjust the learning rate using cosine annealing. See the Line 2 to 9 of the Table 1, in which we set the learning rate, the weight decay, the annealing frequency, the batch size and the maximum number of training iterations (Max iteration) for each subdataset from Meta-Dataset. These results are finalized

$$
\begin{aligned}
\frac{\partial F_k}{\partial \boldsymbol{\theta}} &= \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\theta}} + (\frac{\partial \mathbf{W}_k^*}{\partial \boldsymbol{\theta}})^\top \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \mathbf{W}_k^*} + (\frac{\partial \boldsymbol{\phi}_k^*}{\partial \boldsymbol{\theta}})^\top \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\phi}_k^*} \\
&+ \lambda(\frac{\partial \mathbf{W}_k^*}{\partial \boldsymbol{\theta}})^\top (\frac{\partial g(\mathbf{W}_k^*)}{\partial \mathbf{W}_k^*})^\top [g(\mathbf{W}_k^*) - g(\mathbf{W}_C)] \\
&= \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\theta}} + (\frac{\partial \mathbf{W}_k^*}{\partial \boldsymbol{\theta}})^\top \left\{ \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \mathbf{W}_k^*} + \lambda(\frac{\partial g(\mathbf{W}_k^*)}{\partial \mathbf{W}_k^*})^\top [g(\mathbf{W}_k^*) - g(\mathbf{W}_C)] \right\} \\
&+ (\frac{\partial \boldsymbol{\phi}_k^*}{\partial \boldsymbol{\theta}})^\top \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\phi}_k^*} \\
&\overset{①}{=} \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\theta}} + \mathbf{0} + \mathbf{0} \\
&= \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\theta}},
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
\frac{\partial F_k}{\partial \mathbf{W}_C} &= (\frac{\partial \mathbf{W}_k^*}{\partial \mathbf{W}_C})^\top \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \mathbf{W}_k^*} + (\frac{\partial \boldsymbol{\phi}_k^*}{\partial \mathbf{W}_C})^\top \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \boldsymbol{\phi}_k^*} \\
&+ \lambda[(\frac{\partial g(\mathbf{W}_C)}{\partial \mathbf{W}_C})^\top - (\frac{\partial \mathbf{W}_k^*}{\partial \mathbf{W}_C})^\top (\frac{\partial g(\mathbf{W}_k^*)}{\partial \mathbf{W}_k^*})^\top][g(\mathbf{W}_C) - g(\mathbf{W}_k^*)] \\
&= (\frac{\partial \mathbf{W}_k^*}{\partial \mathbf{W}_C})^\top \left\{ \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \mathbf{W}_k^*} + \lambda(\frac{\partial g(\mathbf{W}_k^*)}{\partial \mathbf{W}_k^*})^\top [g(\mathbf{W}_k^*) - g(\mathbf{W}_C)] \right\} + (\frac{\partial \phi_k^*}{\partial \mathbf{W}_C})^\top \frac{\partial \mathcal{L}_{D_k}(\boldsymbol{\theta}, \mathbf{W}_k^*, \boldsymbol{\phi}_k^*)}{\partial \phi_k^*} \\
&+ \lambda(\frac{\partial g(\mathbf{W}_C)}{\partial \mathbf{W_C}})^\top [g(\mathbf{W}_C) - g(\mathbf{W}_k^*)] \\
&\overset{②}{=} \mathbf{0} + \mathbf{0} + \lambda(\frac{\partial g(\mathbf{W}_C)}{\partial \mathbf{W_C}})^\top [g(\mathbf{W}_C) - g(\mathbf{W}_k^*)] \\
&= \lambda(\frac{\partial g(\mathbf{W}_C)}{\partial \mathbf{W_C}})^\top [g(\mathbf{W}_C) - g(\mathbf{W}_k^*)],
\end{aligned} \tag{4}
$$

by evaluating the performance on the validation set of different values of hyperparameters. Meanwhile, we use data augmentation in the training stage, like random color augmentations and random crops. And for MDL, we need to train a separate model on the eight seen datasets, and the train hyperparameters as mentioned above is shown in the last line of the Table 1.

In addition, we use the same setting in the upper-layer optimization of our model 2L_Meta as MDL, and in the lower layer, the learning rate is consistent with the upper layer at each iteration, and the number of update steps is 2.

3. More results

3.1. Performance of SDLs on Meta-dataset

We use the eight SDLs to perform intra-domain generalization and inter-domain generalization, that is to evaluate each SDL on each subdataset from Meta-Dataset. And We compute and compare the average performance. See the Table 2, and we can find that the optimal model of the specific dataset is the model trained by the dataset or the ImageNet model. We can also see that the SDL trained by ImageNet outperforms other SDLs substantially, which may be due to the large number and variety of images in ImageNet.

3.2. Impact of the subspace dimension on generalization

In our method, the subspace dimension, *i.e.*, the rank m of the basis vector matrix $\mathbf{W}$ is a hyperparameter to be tuned. The results of different m on Meta-Datasets are shown in the Table 3. We find that settings with larger parameter values give better results than smaller values, probably because subspaces with higher dimensioncontain more feature information, while too high value leads to overfitting.We use 384 in our model.

3.3. Impact of gradient update step number in the lower-layer optimization.

Number of steps for gradient update in the lower layer of our meta-problem is set as a hyperparameter n. The results

Table 1. Training hyperparameters of multiple SDLs and MDL on Meta-Dataset. The first column represent different SDLs on different subdataset of Meta-Dataset.

Model	Learning rate	Weight decay	Annealing frequency	Batch size	Max iteration
ImageNet	3×10^{-2}	7×10^{-4}	48000	64	480000
Omniglot	3×10^{-2}	7×10^{-4}	3000	16	5000
Aircraft	3×10^{-2}	7×10^{-4}	3000	8	5000
Birds	3×10^{-2}	7×10^{-4}	3000	16	5000
Textures	3×10^{-2}	7×10^{-4}	1500	32	5000
Quick Draw	1×10^{-2}	7×10^{-4}	48000	64	480000
Fungi	3×10^{-2}	7×10^{-4}	1500	32	480000
VGG Flower	3×10^{-2}	7×10^{-4}	1500	8	5000
MDL	3×10^{-2}	7×10^{-4}	48000	64	240000

Table 2. Results of all SDLs. Mean accuracy and 95reported. All results are obtained during meta-testing phase. The test tasks of the domain corresponding to the trained model for each column belong to in-domain generalization, while the remaining domains are out-of-domain generalization. We also report overall accuracy for all domains.

DatasetModel	ImageNet	Omniglot	Aircraft	Birds	Textures	Quick Draw	Fungi	VGG Flower
ImageNet	**55.78** ± 1.0	16.01 ± 0.5	21.18 ± 0.7	26.26 ± 0.8	26.00 ± 0.8	22.53 ± 0.7	32.30 ± 0.9	24.5 ± 0.8
Omniglot	65.73 ± 1.3	**93.20** $\pm$ **0.5**	56.87 ± 1.3	57.75 ± 1.2	54.37 ± 1.3	77.24 ± 1.0	56.20 ± 1.2	54.0 ± 1.2
Aircraft	49.77 ± 0.9	17.37 ± 0.5	**85.74** $\pm$ **0.5**	29.48 ± 0.7	23.72 ± 0.6	25.48 ± 0.7	31.08 ± 0.7	24.5 ± 0.6
Birds	70.43 ± 0.8	13.51 ± 0.5	19.14 ± 0.7	**71.24** $\pm$ **0.9**	22.80 ± 0.7	17.59 ± 0.7	43.16 ± 0.9	27.4 ± 0.8
Textures	**72.95** $\pm$ **0.6**	29.19 ± 0.5	41.32 ± 0.6	42.73 ± 0.6	60.40 ± 0.7	39.40 ± 0.7	57.27 ± 0.7	42.1 ± 0.7
Quick Draw	55.20 ± 0.9	52.53 ± 0.9	40.22 ± 1.0	38.66 ± 0.9	41.59 ± 1.0	**82.81** ± 0.6	35.57 ± 0.9	39.5 ± 1.1
Fungi	42.72 ± 1.1	9.80 ± 0.5	13.10 ± 0.6	25.70 ± 0.9	17.59 ± 0.8	11.92 ± 0.6	**65.78** $\pm$ **0.9**	23.4 ± 0.7
VGG Flower	**86.96** $\pm$ **0.6**	23.66 ± 0.6	47.03 ± 0.8	63.74 ± 0.8	49.23 ± 0.9	35.32 ± 0.8	79.70 ± 0.7	78.2 ± 0.6
Traffic Sign	**48.28** $\pm$ **1.0**	16.27 ± 0.6	33.92 ± 0.9	35.65 ± 0.9	37.54 ± 1.0	30.79 ± 1.0	26.77 ± 0.7	30.3 ± 0.8
MSCOCO	**51.99** $\pm$ **1.0**	14.41 ± 0.6	20.10 ± 0.7	25.17 ± 0.8	26.57 ± 0.9	19.39 ± 0.8	30.21 ± 0.9	25.6 ± 1.8
MNIST	78.00 ± 0.6	**92.82** $\pm$ **0.4**	67.94 ± 0.7	78.18 ± 0.6	72.39 ± 0.7	87.61 ± 0.5	66.03 ± 0.7	72.9 ± 0.7
CIFAR-10	**69.64** $\pm$ **0.7**	29.89 ± 0.6	38.53 ± 0.7	39.60 ± 0.7	40.78 ± 0.7	38.37 ± 0.7	37.03 ± 0.7	40.3 ± 0.8
CIFAR-100	**58.56** $\pm$ **1.0**	13.56 ± 0.7	23.24 ± 0.8	29.81 ± 0.9	29.12 ± 0.9	23.62 ± 0.9	27.02 ± 0.9	29.0 ± 1.0
Average	61.7	32.5	39.2	43.40	38.62	39.4	45.6	39.4

Table 3. **Quantiative analysis of subspace dimension.** on Meta-Dataset.

Datasetdimension m	256	320	**384**	446
In-domain	72.54	74.31	79.72	77.39
Out-of-domain	53.32	63.54	69.35	68.97
Average	65.15	70.17	**75.73**	74.15

Table 4. **Quantiative analysis of gradient update step number** on Meta-Dataset.

Datasetstep n	1	**2**	3	5
In-domain	79.18	79.72	78.21	77.89
Out-of-domain	68.79	69.35	67.67	66.35
Average	75.18	**75.73**	74.16	73.45

of different n on Meta-Datasets are shown in the Table 4. A good choice for our model is $n = 2$ according to attempts already made

References

[1] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part X*, volume 12355 of *Lecture Notes in Computer Science*, pages 769–786. Springer, 2020. 1

[2] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal repre-

sentation learning from multiple domains for few-shot classification. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9506–9515. IEEE, 2021. 1

[3] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 1