Meta-Transfer Learning for Few-Shot Learning
Supplementary Materials

Qianru Sun[1,3*] Yaoyao Liu[2*] Tat-Seng Chua[1] Bernt Schiele[3]

[1]National University of Singapore [2]Tianjin University[†]
[3]Max Planck Institute for Informatics, Saarland Informatics Campus
{qsun, schiele}@mpi-inf.mpg.de
liuyaoyao@tju.edu.cn {dcssq, dcscts}@nus.edu.sg

These supplementary materials include the details of network architecture (§A), implementation (§B), FC100 dataset splits (§C), standard variance analysis (§D), additional ablation results (§E), and some interpretation of our meta-learned model (§F). In addition, our open-source code is on GitHub[1].

A. Network architectures

In Figure S1, we present the 4CONV architecture for feature extractor Θ, as illustrated in Section 5.1 "Network architecture" of the main paper.

In Figure S2, we present the other architecture – ResNet-12. Figure S2(a) shows the details of a single residual block and Figure S2(b) shows the whole network consisting of four residual blocks and a mean-pooling layer.

The input of Θ is the 3-channel RGB image, and the output is the 512-dimensional feature vector. $a = 0.1$ is set for all leakyReLU activation functions in ResNet-12.

B. Implementation details

For **the phase of DNN training on large-scale data**, the model is trained by Adam optimizer [2]. Its learning rate is initialized as 0.001, and decays to its half every $5k$ iterations until it is lower that 0.0001. We set the keep probability of the dropout as 0.9 and batch-size as 64. The pre-training stops after $10k$ iterations. Note that for the hyperparameter selection, we randomly choose 550 samples each class as the training set, and the rest as validation. After the grid search of hyperparameters, we fix them and mix up all samples (64 classes, 600 samples each class), in order to do the final pre-training. Besides, these pre-training samples are augmented with horizontal flip.

For the **meta-train phase**, we sample 5-class, 1-shot (5-shot or 10-shot) episodes to contain 1 (5 or 10) sample(s) for episode training, and 15 samples for episode test uniformly, following the setting of MAML [1]. The base-learner θ is optimized by batch gradient descent with the learning rate of 0.01. It gets updated with 20 and 60 epochs respectively for 1-shot and 5-shot tasks on the miniImageNet dataset, and 20 epochs for all tasks on the FC100 dataset. The meta-learner, *i.e.*, the parameters of the *SS* operations, is optimized by Adam optimizer [2]. Its learning rate is initialized as 0.001, and decays to the half every $1k$ iterations until 0.0001. The size of meta-batch is set to 2 (tasks) due to the memory limit.

Using our **HT meta-batch strategy**, hard tasks are sampled every time after running 10 meta-batches, *i.e.*, the failure classes used for sampling hard tasks are from 20 tasks. The number of hard task is selected for different settings by validation: 10 and 4 hard tasks respectively for the 1-shot and 5-shot experiments on the miniImageNet dataset; and respectively 20, 10 and 4 hard tasks for the 1-shot, 5-shot and 10-shot experiments on the FC100 dataset.

For the **meta-test phase**, we sample 5-class, 1-shot (5-shot or 10-shot) episodes and each episode contains 1 (5 or 10) sample(s) for both episode train and episode test. On each dataset, we sample 600 meta-test tasks. All these settings are exactly the same as MAML [1].

C. Super-class splits on FC100

In this section, we show the details of the FC100 splits according to the super-class labels, same with TADAM [3]. **Training split** super-class indexes: 1, 2, 3, 4, 5, 6, 9, 10, 15, 17, 18, 19; and corresponding labels: fish, flowers, food_containers, fruit_and_vegetables, household_electrical_devices, household_furniture, large_man-made_outdoor_things, large_natural_outdoor_scenes, reptiles, trees, vehicles_1, vehicles_2.

*Equal contribution.
†Yaoyao Liu did this work during his internship at NUS.
[1]https://github.com/y2l/meta-transfer-learning-tensorflow

Validation split super-class indexes: 8, 11, 13, 16; and corresponding labels: large_carnivores, large_omnivores_and_herbivores, non-insect_invertebrates, small_mammals.

Test split super-class indexes: 0, 7, 12, 14; and corresponding labels: aquatic_mammals, insects, medium_mammals, people.

An episode (task) is independently sampled from a corresponding split, *e.g*. a meta-train episode contains 5 classes that can only be belonging to the 12 super-classes in the training split. Therefore, there is no fine-grained information overlap between meta-train and meta-test tasks.

D. Standard variance analysis

The final accuracy results reported in our main paper are the mean values and standard variances of the results of 600 meta-test tasks. The standard variance is affected by the number of episode test samples. As introduced in §B, we use the same setting as MAML [1] which used a smaller number of samples for episode test (1 sample for 1-shot episode test and 5 samples for 5-shot), making the result variance higher. Other works that used more samples for episode test got lower variances, *e.g*., TADAM [3] used 100 samples and its variances are about $\frac{1}{6}$ and $\frac{1}{3}$ of MAML's respectively for miniImageNet 1-shot and 5-shot.

In order to have a fair comparison with TADAM in terms of this issue, we supplement the experiments using 100 episode test samples at the meta-test. We get the new confidence intervals (using our method: MTL w/o HT meta-batch) as 0.71% (0.3% for TADAM) and 0.54% (0.3% for TADAM) respectively for 1-shot and 5-shot on the miniImageNet dataset, and 0.70% (0.4% for TADAM), 0.63% (0.4% for TADAM) and 0.58% (0.5% for TADAM) respectively for 1-shot, 5-shot and 10-shot on the FC100 dataset.

E. Additional ablation study

We supplement the results in Table S1, for the comparisons mentioned in Section 5.1 of main paper. Red numbers on the bottom row are copied from the main paper (corresponding to the MTL setting: *SS* Θ, meta-batch) and shown here for the convenience of comparison.

To get the first row, we train 4CONV net by large-scale data (same to the pre-training of ResNet-12) and get inferior results, as we declared in the main paper. Results on the second and third rows show the performance drop when changing the single FC layer θ to multiple layers, *e.g*. 2 FC layers and 3 FC layers. Results on the fourth row show the performance drop when updating both Θ and θ for the base-learning. The reason is that Θ has too many parameters to update with too little data.

F. Interpretation of meta-learned *SS*

In Figure S3, we show the statistic histograms of learned *SS* parameters, taking miniImageNet 1-shot as an example setting. Scaling parameters Φ_{S_1} are initialized as 1 and shifting parameters Φ_{S_1} as 0. After meta-train, we observe that these statistics are close to Gaussian distributions respectively with $(0.9962,\ 0.0084)$ and $(0.0003,\ 0.0002)$ as (mean, variance) values, which shows that the uniform initialization has been changed to Gaussian distribution through few-shot learning. Possible interpretations are in three-fold: 1) majority patterns trained by a large number of few-shot tasks are close to the ones trained by large-scale data; 2) tail patterns with clear scale and shift values are the ones really contributing to adapting the model to few-shot tasks; 3) tail patterns are of small quantity, enabling the fast learning convergence.

References

[1] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 2

[2] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 1412.6980, 2014. 1

[3] B. N. Oreshkin, P. Rodríguez, and A. Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 1, 2

Meta-learning	Base-learning	FC dim of θ	Feature extractor	miniImageNet		FC100		
				1-shot	5-shot	1-shot	5-shot	10-shot
Φ_{S_1}, Φ_{S_2}	θ	5	4 CONV (pre)	45.6 ± 1.8	61.2 ± 0.9	38.0 ± 1.6	46.4 ± 0.9	56.5 ± 0.8
Φ_{S_1}, Φ_{S_2}	θ (2-layer)	512, 5	ResNet-12 (pre)	59.1 ± 1.9	70.7 ± 0.9	40.3 ± 1.9	53.3 ± 0.9	54.1 ± 0.8
Φ_{S_1}, Φ_{S_2}	θ (3-layer)	1024, 512, 5	ResNet-12 (pre)	56.2 ± 1.8	68.7 ± 0.9	40.0 ± 1.8	52.3 ± 1.0	53.8 ± 0.8
Φ_{S_1}, Φ_{S_2}	Θ, θ	5	ResNet-12 (pre)	59.6 ± 1.8	71.6 ± 0.9	43.3 ± 1.9	54.6 ± 1.0	60.7 ± 0.8
Φ_{S_1}, Φ_{S_2}	θ	5	ResNet-12 (pre)	60.2 ± 1.8	74.3 ± 0.9	43.6 ± 1.8	55.4 ± 0.9	62.4 ± 0.8

Table S1. Additional ablative study. On the last row, we show the red numbers which are reported in our main paper (corresponding to the MTL setting: *SS* $[\Theta; \theta]$, meta-batch).

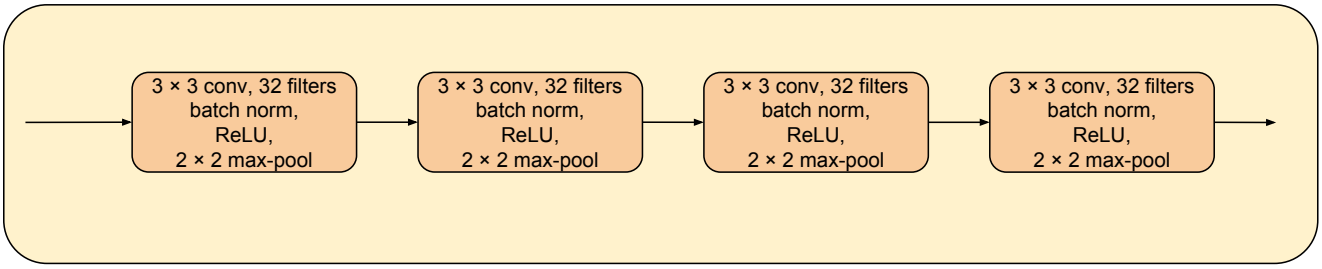

Figure S1. Network architecture of 4CONV

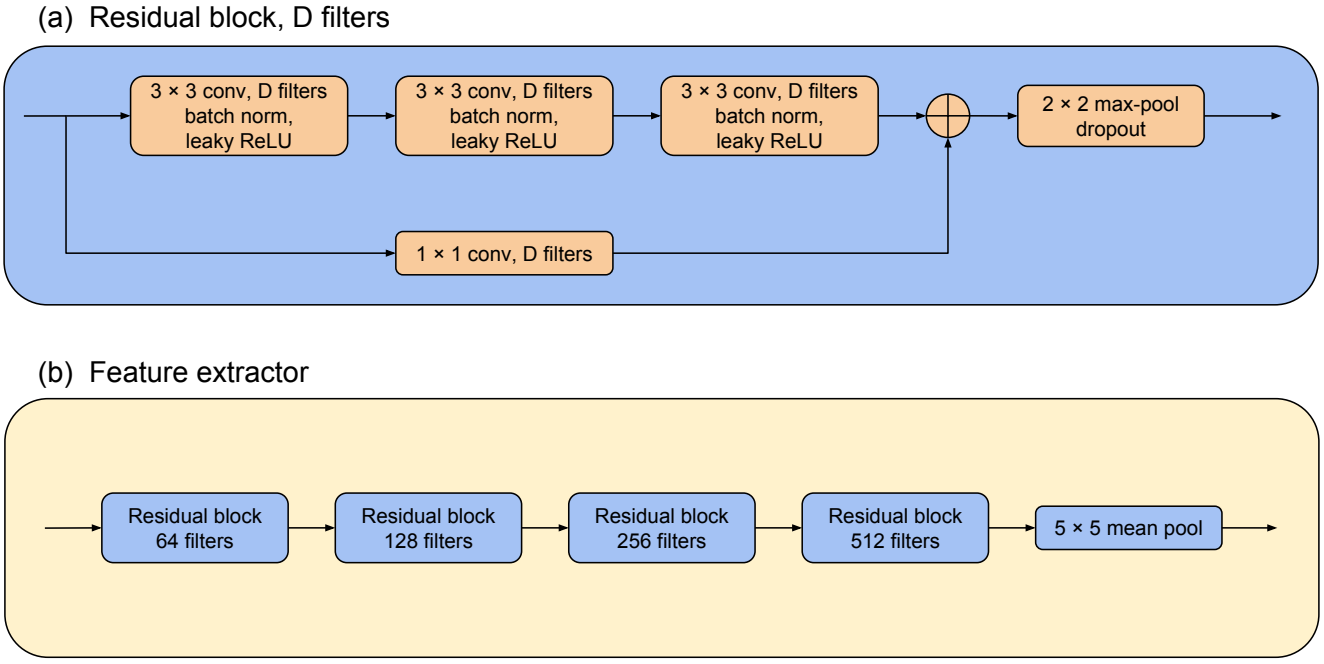

Figure S2. Network architecture of ResNet-12

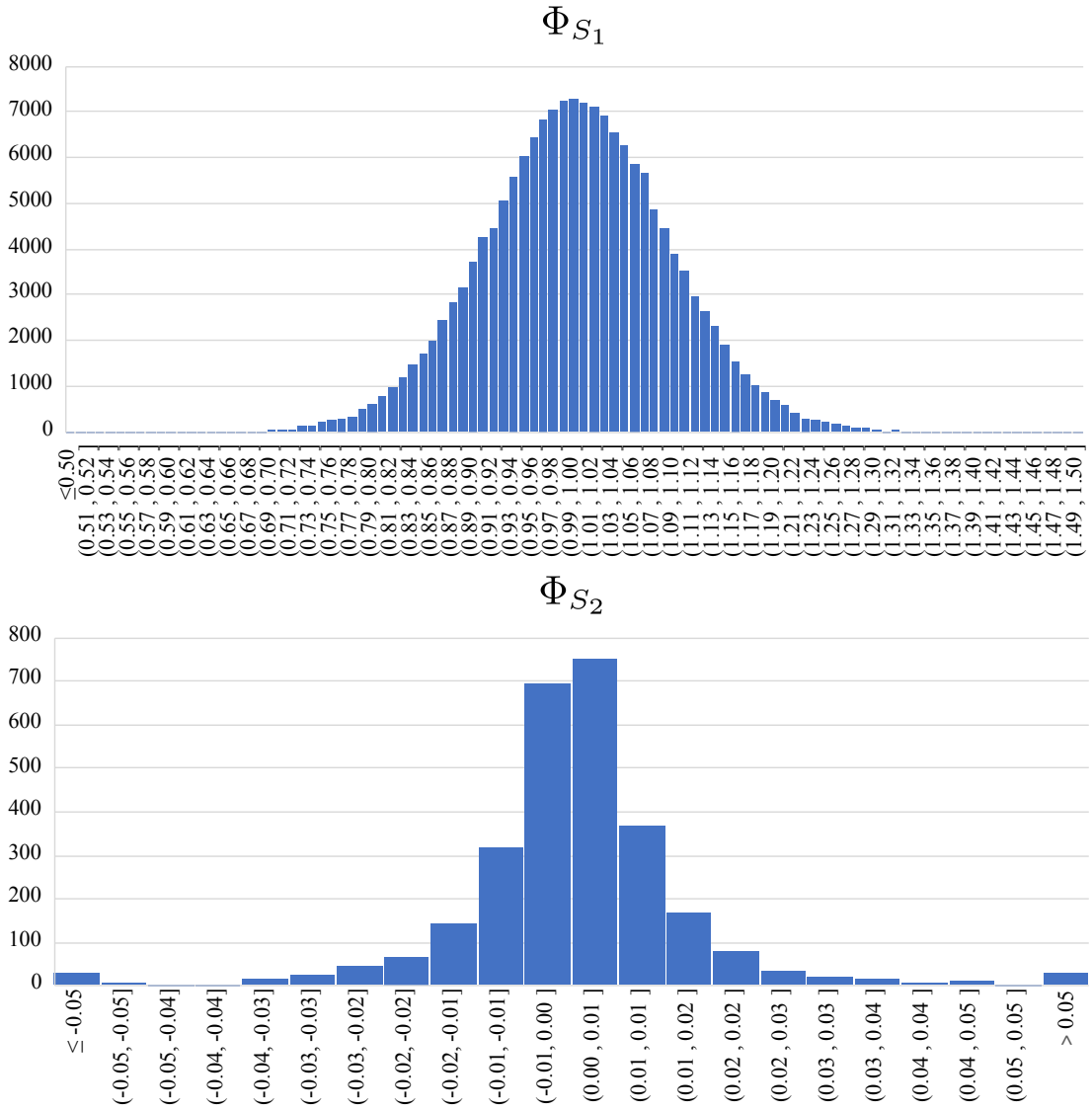

Figure S3. The statistic histograms of learned *SS* parameters, taking miniImageNet 1-shot as an example setting.