Appendices

A. Implementation Details

Implementation details for SS. For SS scenario, all pre-trained models are with the same architecture. We take Conv4 as the architecture of all pre-trained models and the meta model, the same as regular meta-learning works [6, 9, 22, 39]. Conv4 is a four-block convolution neural network, where each block consists of 32 3×3 filters, a BatchNorm, a ReLU and an additional 2×2 max-pooling. The episode-batch size for each iteration is 4. We adopt the Adam optimizer to learn the dynamic dataset, base model and meta model with a learning rate of 0.25, 0.01 and 0.001, respectively. We take one-step gradient descent to perform fast adaptation for both meta training and testing. After 4000 iterations (outer loops), we add curriculum mechanism to episode inversion. At each iteration, the meta model sends a positive feedback if the sum of training accuracy over the episode batch has not increased for 6 consecutive iterations. We empirically set the scaling factor λ as 10. For meta testing, we calibrate the backbone for 1 iteration via Adam optimizer with a learning rate of $1e-5$. We set the temperature parameter as 0.1. Then we freeze the backbone and train a new classifier over the entire support set for 100 iterations via Adam optimizer with a learning rate of 0.01. For episode inversion, we set $\alpha_{TV} = 1e-4$ and $\alpha_{l_2} = 1e-5$.

Implementation details for SH. For SH scenario, all pre-trained models are trained on the same dataset but with heterogeneous architectures. For each task, we pre-train the model with an architecture randomly selected from Conv4, ResNet-10 and ResNet-18. Compared to Conv4, ResNet-10 and ResNet-18 are larger-scale neural networks. The ResNet-10 has 4 residual stages of 1 block each which gradually decreases the spatial resolution. The ResNet-18 also has 4 residual stages but with 2 blocks per stage, which is a deeper neural network. The other configurations are the same as SS.

Implementation details for MH. For MH scenario, all pre-trained models are trained on multiple datasets with heterogeneous architectures. During meta training, we construct each task from the meta training dataset randomly selected from CIFAR-FS and MiniImageNet. We pre-train the model with an architecture randomly selected from Conv4, ResNet-10 and ResNet-18. We take Conv4 as the meta model architecture. During meta testing, we report the average accuracy over 600 tasks sampled from both CIFAR-FS and MiniImageNet meta testing datasets. The other configurations are the same as SS.

Table 7. Hyperparameter sensitivity on DFML CIFAR-FS 5-way classification in SS scenario.

λ	5-way 1-shot	5-way 5-shot
λ=10.0	38.66 ± 0.78	51.95 ± 0.79
λ=2.0	37.87 ± 0.70	49.09 ± 0.75
λ=0.5	38.04 ± 0.79	48.91 ± 0.75

Table 8. Effect of the number of pre-trained models in SS scenario.

num	5-way 1-shot	5-way 5-shot
2	34.28 ± 0.75	45.40 ± 0.74
6	36.78 ± 0.75	47.71 ± 0.80
13	38.66 ± 0.78	51.95 ± 0.79

B. More Results

Hyperparameter Sensitivity. We evaluate the model performance sensitivity with different values λ in Tab. 7. As can be seen, the achieved performance of our method is stable with the changes of λ value, although there are some variations among different λ values. This advantage makes it easy to apply our method in practice.

Effect of the number of pre-trained models. We perform the experiments about DFML with the different numbers of pre-trained models in SS scenario. We train each pre-trained model for a 5-way classification problem. Tab. 8 shows the results. By increasing the number from 2 to 6, we can observe 2.5% and 2.31% performance gains for 1-shot and 5-shot learning, respectively. The gains increase to 4.38 and 6.55% by increasing the number from 2 to 13. The reason is that more pre-trained models can provide more underlying data knowledge of different classes. With broader data knowledge, meta-learning can acquire better generalization for target tasks with new unseen classes.