Supplementary Materials

These supplementary materials include E^{3}BM algorithms, results with confidence intervals, the supplementary plots to Fig. 4, backbone architectures, implementation details, ablation results for MAML, the inference time and the number of parameters, and the execution steps of our source code with PyTorch.

A E^{3}BM algorithms

Algorithm 1 summarizes the meta-training (line 1-10) and meta-testing (line 11-16) procedures in our E^{3}BM approach. For clarity, the base-learning steps within a single episode are moved to Algorithm 2.

Algorithm 1: An Ensemble of Epoch-wise Empirical Bayes Models (E^{3}BM)

Input: Meta-train episode distribution $p_{tr}(\mathcal{T})$, Meta-test episode distribution $p_{te}(\mathcal{T})$, and meta-train stepsizes β_1 and β_2.
Output: The average accuracy of meta-test.
% Meta-train phase:
1 Randomly initialize θ;
2 **for** *all meta iterations* **do**
3   Sample a batch of meta-train episodes $\{\mathcal{T}_i\} \in p_{tr}(\mathcal{T})$;
4   **for** $\mathcal{T}_i$ *in* $\{\mathcal{T}_i\}$ **do**
5   Train the sequence of base-learners on $\mathcal{T}_i$ by **Algorithm** 2;
6   **end**
7   Evaluate $\mathcal{L}^{(te)}$ with Eq. (13) ;
8   Optimize Ψ_α, and Ψ_v with Eq. (14) using β_1 and β_2;
9   Optimize other meta components, e.g., θ.
10 **end**
% Meta-test phase:
11 Sample meta-test episodes $\{\mathcal{T}_i\} \in p_{te}(\mathcal{T})$;
12 **for** $\mathcal{T}_i$ *in* $\{\mathcal{T}_i\}$ **do**
13   Train the sequence of base-learners on $\mathcal{T}_i$ and obtain the prediction scores $\hat{y}^{(te)}$ by **Algorithm** 2;
14   Compute episode test accuracy Acc_i;
15 **end**
16 Return the average accuracy of $\{Acc_i\}$.

B Results with confidence intervals

In Table S1, we supplement the few-shot classification accuracy (%) on *mini*ImageNet, *tiered*ImageNet, and FC100 (5-class) with confidence intervals.

Algorithm 2: Learning the ensemble of base-learners in one episode

```
Input: An episode 𝒯, hyperprior learners Ψ_α and Ψ_v.
Output: Prediction ŷ^(te), and episode test loss ℒ^(te).
Initialize Θ_0 = θ;
for m in {1, ..., M} do
    Evaluate ℒ_m^(tr) with Eq. (7) and compute ∇_Θ ℒ_m^(tr);
    Get α_m from Ψ_α and get v_m from Ψ_v;
    Get Θ_m using α_m with Eq. (6);
    Compute z_m with Eq. (8);
    if m = 1 then
        Initialize ŷ_1^(te) = v_1 z_1;
    else
        Compute ŷ_m^(te) using v_m with Eq. (9);
    end
end
Evaluate ℒ^(te) with Eq. (13).
```

Methods	Backbone	*mini*ImageNet		*tiered*ImageNet		FC100	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML+E^{3}BM	4CONV	53.2 ± 1.8	65.1 ± 0.9	52.1 ± 1.8	70.2 ± 0.9	39.9 ± 1.8	52.6 ± 0.9
MTL+E^{3}BM	ResNet-12	63.8 ± 0.4	80.1 ± 0.3	71.2 ± 0.4	85.3 ± 0.3	43.2 ± 0.3	60.2 ± 0.3
MTL+E^{3}BM	ResNet-25	64.3 ± 0.4	81.0 ± 0.3	70.0 ± 0.4	85.0 ± 0.3	45.0 ± 0.4	60.5 ± 0.3
SIB+E^{3}BM	WRN-28-10	71.4 ± 0.5	81.2 ± 0.4	75.6 ± 0.6	84.3 ± 0.4	46.0 ± 0.6	57.1 ± 0.4

Table S1. Supplementary to Table 1. Few-shot classification accuracy (%) on *mini*ImageNet, *tiered*ImageNet, and FC100 (5-class).

C Supplementary figures

Supplementary to Fig. 4(a)(b). In Fig. S1, we supplement the meta-validation accuracies for the 1-shot and 5-shot cases on *mini*ImageNet, *tiered*ImageNet, and FC100 (Note that Fig. 4 already has the *mini*ImageNet 1-shot results).

Supplementary to Fig. 4(c)(d). On the *mini*ImageNet, we supplement the plots of α and v in the 5-shot case in Fig. S2(a)(b). On the *tiered*ImageNet, we show the plots of α and v in Fig. S2(c)(d) and (e)(f), respectively for 1-shot and 5-shot cases. On the FC100, we show the plots of α and v in Fig. S2(g)(h) and (i)(j), respectively for 1-shot and 5-shot cases. Each figure demonstrates the values of α (or v) generated by the model "MTL+E^{3}BM" as in Table 1.

D Backbone architectures

4CONV consists of 4 layers with 3×3 convolutions and 32 filters, followed by batch normalization (BN), a ReLU nonlinearity, and 2×2 max-pooling.

ResNet-12 has 3 residual blocks. Each block has 4 convolution layers with 3×3 kernels. The number of filters starts from 160 and is doubled every next block. After a global average pooling layer, it gets a 640-dim embedding. This architecture follows [37].

ResNet-25 has 3 residual blocks after an initial convolution layer. Each block has 8 convolution layers with 3×3 kernels. The number of filters starts from 160 and is doubled every next block. After a global average pooling layer, it gets a 640-dim embedding. This architecture follows [78].

WRN-28-10 has its depth and width set to 28 and 10, respectively. After a global average pooling in the last layer of the backbone, it gets a 640-dimensional embedding. For this backbone, we resize the input image to $80 \times 80 \times 3$ for a fair comparison with related methods [25, 70]. Other details are the same as those with ResNet-25 [61, 78].

E Implementation details

MTL+E^{3}BM. The meta learning rates for the scaling and shifting weights Φ_{SS} and the base-learner initializer θ are set to 1×10^{-4} uniformly. The base learning rates $\{\alpha'_m\}_{m=1}^{M}$ (Fig. 3) are initialized as 1×10^{-2} [70,78]. We meta-train MTL+E^{3}BM for $10,000$ iterations and use the model, which has the highest meta-validation accuracy, for meta-test.

SIB+E^{3}BM. The meta learning rates for both SIB network $\phi(\lambda, \xi)$ and base-learner initializer θ are set to 1×10^{-3} uniformly. The base learning rates $\{\alpha'_m\}_{m=1}^{M}$ (Fig. 3) are initialized as 1×10^{-3} [25]. We meta-train SIB+E^{3}BM for $50,000$ iterations and use the model, which has the highest meta-validation accuracy, for meta-test.

MAML+E^{3}BM. MAML only contains a model initializer θ, and we set its meta-learning rate as 1×10^{-3} [13]. The base learning rates $\{\alpha'_m\}_{m=1}^{M}$ (Fig. 3) are initialized as 1×10^{-3}. We meta-train MAML+E^{3}BM for $60,000$ iterations and use the model, which has the highest meta-validation accuracy, for meta-test.

Shared hyperparameters. The meta learning rates for Ψ_α and Ψ_v are set to 1×10^{-6} uniformly. For initializing $\{v'_m\}_{m=1}^{M}$ (Fig. 3), we have two options. One is each v'_m is initialized as 1/(number of base-learners), and the other one is that $\{v'_m\}_{m=1}^{M-1}$ are initialized as 0 and v'_M as 1. In Eq. (12) in Sec. 4.3, λ_1 and λ_2 are set to 1×10^{-4}. For the rest of the hyperparameters, we follow the original settings of baselines [13, 25, 70].

Constraints for v and α. In the constraint mode, we applied the constraints on v and α to force them to be positive and smaller than 1. We did not have any constraint for Δv or $\Delta \alpha$. Please note that the constraints are not applied in the default setting.

Dataloader For MAML, we use the same dataloader as [13]. For MTL, we follow [78, 81, 82]. For SIB, we follow [25].

F Ablation results for MAML

In Table S2, we supplement the ablation results for "MAML+E^{3}BM" on *mini*ImageNet, *tiered*ImageNet, and FC100 (5-class).

No.	**Setting**			***mini*ImageNet**		***tiered*ImageNet**		**FC100**	
	Method	Hyperprior	Learning	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
1	MAML [13]	–	Ind.	48.70	63.11	49.0	66.5	38.1	50.4
2	5x MAML	–	Ind.	52.1	65.1	51.1	68.8	40.1	50.8
3	MAML+E^{3}BM	FC	Ind.	52.1	65.1	51.1	68.8	39.5	51.7
4	MAML+E^{3}BM	FC	Tra.	52.8	65.3	52.2	69.5	40.4	51.8
5	MAML+E^{3}BM	LSTM	Ind.	53.2	65.0	52.1	70.2	39.9	52.6
6	MAML+E^{3}BM	LSTM	Tra.	53.8	65.2	52.7	70.5	40.4	52.3

Table S2. Supplementary to Table 2. Results (%) for different hyperprior learners on *mini*ImageNet, *tiered*ImageNet, and FC100 (5-class). "Ind." and "Tra." denote inductive and transductive settings, respectively.

G The inference time and the number of parameters

In Table S3, we supplement the the inference time and the number of parameters of baselines (100 epochs, *mini*ImageNet, 5-way 1-shot, on NVIDIA V100 GPU)

No.	**Method**	**Backbone**	**# Param**	**Time (min)**
1	MTL	ResNet-25	4,321k	73.3
2	MTL+E^{3}BM (ours)	ResNet-25	4,351k	77.6
3	SIB	WRN-28-10	36,475k	325.0
4	SIB+E^{3}BM (ours)	WRN-28-10	36,490k	331.8
5	ProtoNets	ResNet-12	12,424k	35.2
6	MatchNets	ResNet-12	12,424k	37.3
7	ProtoNets	ResNet-25	36,579k	70.5
7	ProtoNets	WRN-28-10	36,482k	350.1

Table S3. Supplementary to Table 1. The inference time and the number of parameters of baselines (100 epochs, *mini*ImageNet, 5-way 1-shot, on NVIDIA V100 GPU).

H Executing the source code with PyTorch

We provide our PyTorch code at https://gitlab.mpi-klsb.mpg.de/yaoyaoliu/e3bm. To run this repository, we kindly advise you to install python 3.6 and PyTorch

1.2.0 with Anaconda. You may download Anaconda and read the installation instruction on the official website (https://www.anaconda.com/download/). Create a new environment and install PyTorch and torchvision on it:

```
conda create --name e3bm-pytorch python=3.6
conda activate e3bm-pytorch
conda install pytorch=1.2.0
conda install torchvision -c pytorch
```

Install other requirements:

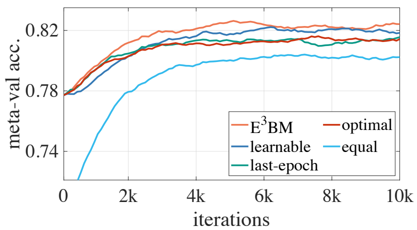
```
pip install -r requirements.txt
```

Run meta-training with default settings (data and pre-trained model will be downloaded automatically):

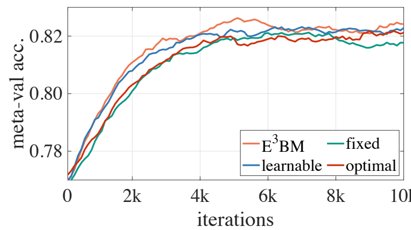
```
python main.py -backbone resnet12 -shot 1 -way 5 -mode meta_train -dataset miniimagenet
python main.py -backbone resnet12 -shot 5 -way 5 -mode meta_train -dataset miniimagenet
python main.py -backbone resnet12 -shot 1 -way 5 -mode meta_train -dataset tieredimagenet
python main.py -backbone resnet12 -shot 5 -way 5 -mode meta_train -dataset tieredimagenet
```

Run pre-training with default settings:

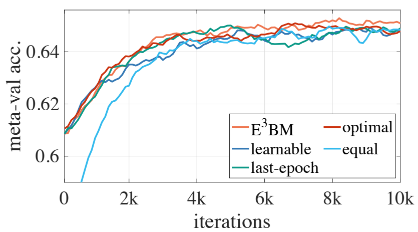
```
python main.py -backbone resnet12 -mode pre_train -dataset miniimagenet
python main.py -backbone resnet12 -mode pre_train -dataset tieredimagenet
```



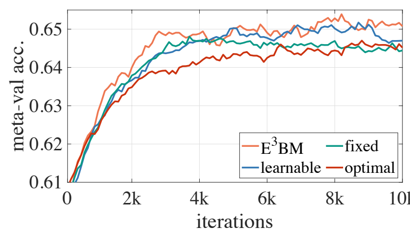
(a) v, *mini*ImageNet, 5-shot



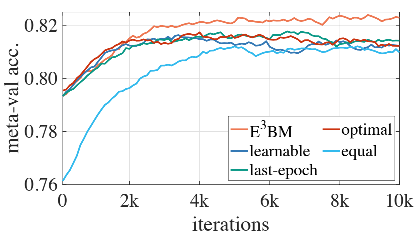
(b) α, *mini*ImageNet, 5-shot



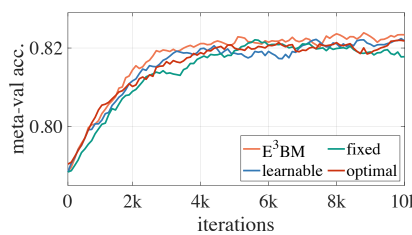
(c) v, *tiered*ImageNet, 1-shot



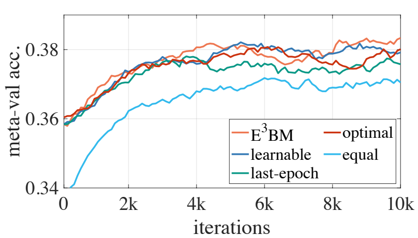
(d) α, *tiered*ImageNet, 1-shot



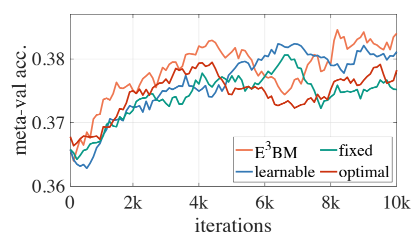
(e) v, *tiered*ImageNet, 5-shot



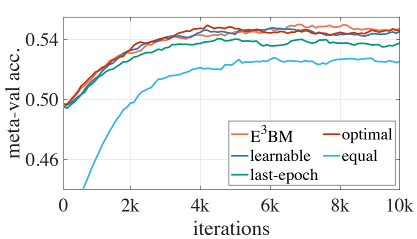
(f) α, *tiered*ImageNet, 5-shot



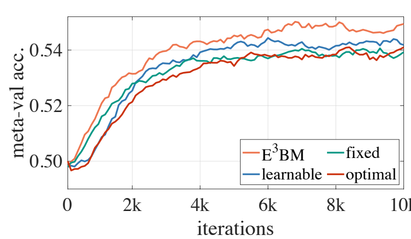
(g) v, FC100, 1-shot



(h) α, FC100, 1-shot

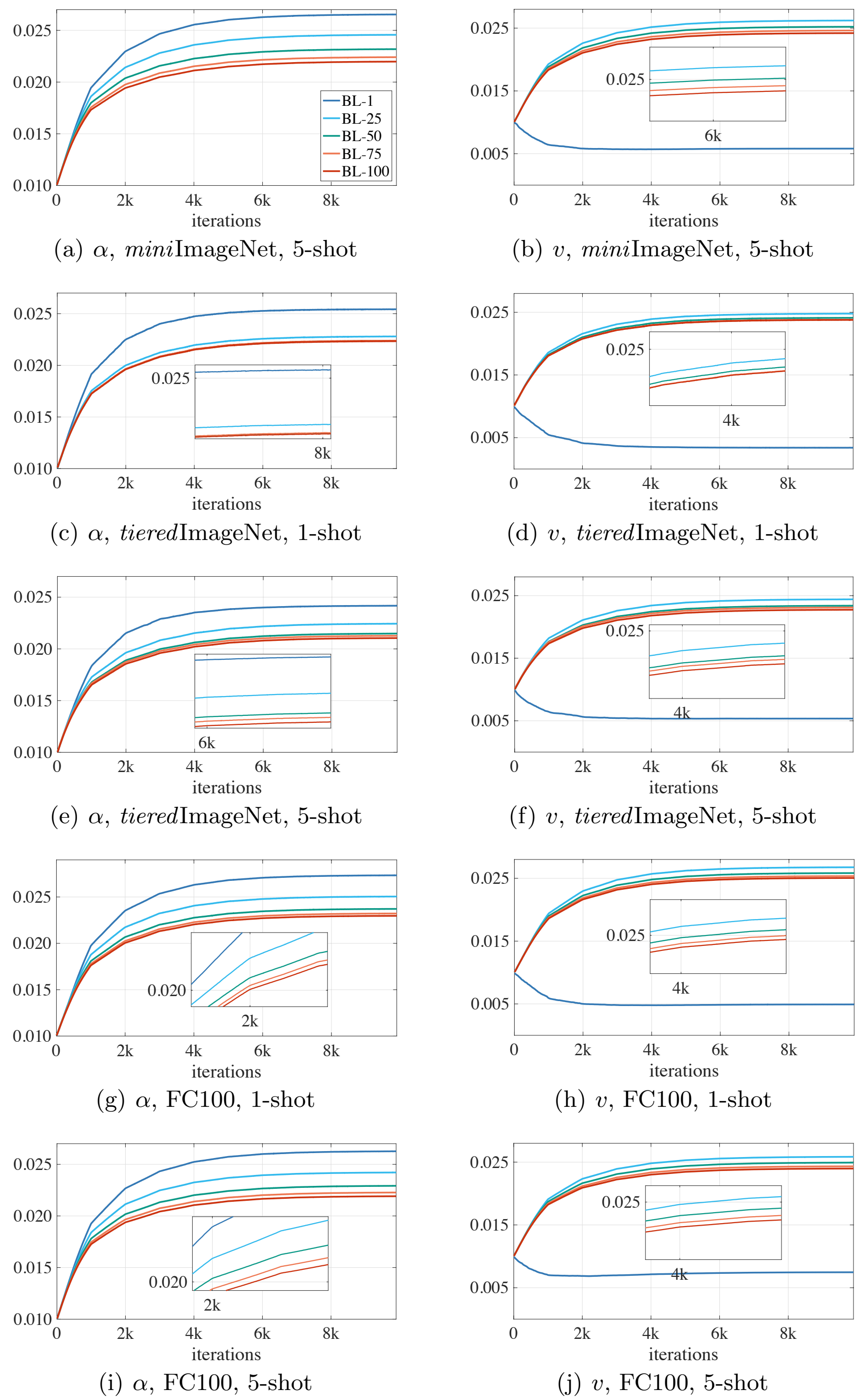


(i) v, FC100, 5-shot



(j) α, FC100, 5-shot

Fig. S1. Supplementary to Fig. 4(a)(b). The meta-validation accuracies of ablation models. Each figure demonstrates the results using the same model "MTL+E^{3}BM" as in Table 1. All curves are smoothed with a rate of 0.9 for a better visualization.



(a) α, *mini*ImageNet, 5-shot
(b) v, *mini*ImageNet, 5-shot
(c) α, *tiered*ImageNet, 1-shot
(d) v, *tiered*ImageNet, 1-shot
(e) α, *tiered*ImageNet, 5-shot
(f) v, *tiered*ImageNet, 5-shot
(g) α, FC100, 1-shot
(h) v, FC100, 1-shot
(i) α, FC100, 5-shot
(j) v, FC100, 5-shot

Fig. S2. Supplementary to Fig. 4(c)(d). The values of α and v generated by Ψ_α and Ψ_v, respectively. Each figure demonstrates the results using the same model "MTL+E^{3}BM" as in Table 1.