Hierarchically Structured Meta-learning: Supplementary Material

Huaxiu Yao[† 1] **Ying Wei**[2] **Junzhou Huang**[2] **Zhenhui Li**[1]

A. Detailed Theoretical Analysis

Proof of Theorem 1 Assuming a task $\mathcal{T}_i$ is sampled from $\mathcal{E}$, its training and testing samples are i.i.d. drawn from distribution $\mathcal{S}_i$, i.e., $\mathcal{D}^{tr}_{\mathcal{T}_i} \sim \mathcal{S}_i$ and $\mathcal{D}^{te}_{\mathcal{T}_i} \sim \mathcal{S}_i$. According to Theorem 3 in (Kuzborskij & Lampert, 2017), if $\mathcal{L}$ is convex, the base learner $f_{\theta_{\mathcal{T}_i}}$ SGD is $\epsilon(\mathcal{S}_i, \theta_0)$-on-average-stable with

$$\epsilon(\mathcal{S}_i, \theta_0) = \mathcal{O}\bigg(\sqrt{c(R(\theta_0) - R^*)}\frac{\sqrt[4]{T}}{n^{tr}} + c\sigma\frac{\sqrt{T}}{n^{tr}}\bigg), \tag{1}$$

where $R^* = \inf_{\theta\in\mathcal{H}} R(\theta)$.

For a new task $\mathcal{T}_t$, we first prove that the initialization can be approximately represented as $\theta_{0t} = \sum_{k=1}^{K} \hat{\mathbf{B}}_k\theta_0$. Wihtout loss of generality, here we consider a hierarchy $C - L - 1$ in HSML.

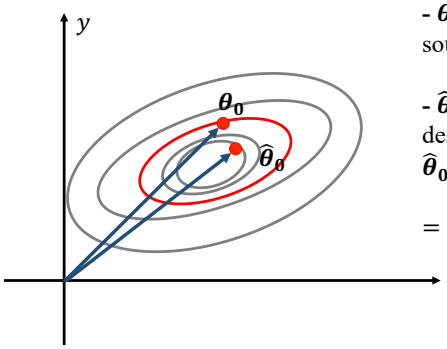
$$\begin{aligned}
\theta_{0t} &= \theta_0 \circ \mathbf{o}_t \\
&= \text{diag}(\mathbf{o}_t)\theta_0 \\
&= \text{diag}(\text{FC}^{\sigma}_{\mathbf{W}_g}(\mathbf{g}_t \oplus \mathbf{h}_t))\theta_0 \\
&= \text{diag}(\text{FC}^{\sigma}_{\mathbf{W}_g}(\mathbf{g}_t \oplus \mathbf{h}_t))\theta_0 \\
&\approx \text{diag}\{a_1[\mathbf{W}_g(\mathbf{g}_t \oplus \mathbf{h}_t)] + a_2\}\theta_0 \\
&= \text{diag}[\mathbf{W}'_g(\mathbf{g}_t \oplus \mathbf{h}_t) + a_2]\theta_0 \\
&= \text{diag}\bigg\{\mathbf{W}'_{gg}\mathbf{g}_t \oplus \mathbf{W}'_{gh}\sum_{l=1}^{L} p^l \tanh\bigg[\mathbf{W}\big(\sum_{c=1}^{C} p^{cl}\tanh(\mathbf{W}^l\mathbf{h}^c_t + b^l)\big) + b\bigg] + a_2\bigg\} \\
&\approx \text{diag}\bigg\{\mathbf{W}'_{gg}\mathbf{g}_t \oplus \mathbf{W}'_{gh}\sum_{l=1}^{L} p^l \bigg[\mathbf{W}\big(\sum_{c=1}^{C} p^{cl}(\mathbf{W}^l\mathbf{h}^c_t + b^l)\big) + b\bigg] + a_2\bigg\} \\
&= \text{diag}\bigg\{\sum_{l=1}^{L}\sum_{c=1}^{C}\bigg[\frac{1}{LC}\mathbf{W}'_{gg}\mathbf{g}_t \oplus p^l\mathbf{W}'_{gh}\bigg(p^{cl}\mathbf{W}\mathbf{W}^l\mathbf{h}^c_t + p^{cl}\mathbf{W}b^l + \frac{b}{C}\bigg) + \frac{a_2}{LC}\bigg]\bigg\}\theta_0 \\
&= \sum_{k=1}^{K}\hat{\mathbf{B}}_k\theta_0,
\end{aligned} \tag{2}$$

where $K = CL$ and $\hat{\mathbf{B}}_{(l-1)*C+c} = \frac{1}{LC}\mathbf{W}'_{gg}\mathbf{g}_t \oplus p^l\mathbf{W}'_{gh}\bigg(p^{cl}\mathbf{W}\mathbf{W}^l\mathbf{h}^c_t + p^{cl}\mathbf{W}b^l + \frac{b}{C}\bigg) + \frac{a_2}{LC}$. Note that the first equality holds by converting the Hadamard product into matrix multiplication, and the first and the second approximations come from first-order taylor series of sigmoid and hybolic functions. In addition, in the $C - L - 1$ hierarchical structure, $\forall l$, $p^l = 1$.

From Eqn. 1, we can see that $\epsilon(\mathcal{S}_t, \theta_0)$ depends $\sqrt{R(\theta_0)}$. Like (Kuzborskij & Lampert, 2017), when the optimization process for task $\mathcal{T}_t$ starts from the equivalent form that $\theta_{0t} = \sum_{k=1}^{K}\hat{\mathbf{B}}_k\theta_0$, we can bound $\epsilon(\mathcal{S}_t, \theta_{0t})$ by using Hoeffding

[†]Part of the work was done when the author interned in Tencent AI Lab. [1]College of Information Science and Technology, Pennsylvania State University, PA, USA [2]Tencent AI Lab, Shenzhen, China. Correspondence to: Ying Wei <judyweiying@gmail.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.



- $\boldsymbol{\theta_0}$: Transferred initialization from source tasks

- $\widehat{\boldsymbol{\theta}}_{\boldsymbol{0}}$: update from one step gradient descent, then

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{0}} = \left(\boldsymbol{I} - \boldsymbol{\alpha}\boldsymbol{\nabla}\boldsymbol{\mathcal{L}}(\boldsymbol{\theta_0})(\boldsymbol{\theta_0 I})^{-1}\right)\boldsymbol{\theta_0} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \widehat{\boldsymbol{B}}_{\boldsymbol{k}}\boldsymbol{\theta_0}$$

Figure 1. Illustration of Existance of $\sum_{k=1}^{K} \hat{\mathbf{B}}_k$.

bound as:

$$\epsilon(\mathcal{S}_t, \theta_{0t}) \leq \mathcal{O}\left(\sqrt{\hat{R}_{\mathcal{D}^{tr}_{\mathcal{T}_t}}(\theta_{0t}) + \sqrt{\frac{1}{n^{tr}}}}\right). \tag{3}$$

Thus, we reach the conclusion.

Proof of Theorem 2 In non-convex case, we assume $\mathcal{L}$ is η-smooth and has ρ-Lipschitz Hessian. According to the Corollary 1 and Proposition 1 in (Kuzborskij & Lampert, 2017), for task $\mathcal{T}_t$, we define:

$$\gamma = \mathcal{O}\left(\mathbb{E}_{(\mathbf{x}_{t,j},\mathbf{y}_{t,j})\sim\mathcal{D}^{tr}_{\mathcal{T}_t}}[\|\nabla^2\mathcal{L}(\theta_{0t},(\mathbf{x}_{t,j},\mathbf{y}_{t,j}))\|_2] + \sqrt{R(\theta_{0t})}\right), \tag{4}$$

and

$$\hat{\gamma} = \frac{1}{n^{tr}}\sum_{j=1}^{n^{tr}}\|\nabla^2\mathcal{L}(\theta_{0t},(\mathbf{x}_{t,j},\mathbf{y}_{t,j}))\|_2 + \sqrt{\hat{R}_{\mathcal{D}^{tr}_{\mathcal{T}_t}}(\theta_{0t})}. \tag{5}$$

Then, we use Hoeffding inequality and get

$$|\gamma - \hat{\gamma}| \leq \mathcal{O}(\frac{1}{\sqrt[4]{n^{tr}}}). \tag{6}$$

Finally, let $\hat{\gamma}^{\pm} = \hat{\gamma} \pm 1/\sqrt[4]{n^{tr}}$, $\epsilon(\mathcal{S}_t, \theta_{0t})$ can be bounded as:

$$\epsilon(\mathcal{S}_t, \theta_{0t}) \leq \mathcal{O}\left(\left(1 + \frac{1}{c\hat{\gamma}^-}\right)\hat{R}_{\mathcal{D}^{tr}_{\mathcal{T}_t}}(\theta_{0t})^{\frac{c\hat{\gamma}^+}{1+c\hat{\gamma}^+}}\frac{1}{(n^{tr})^{\frac{1}{1+c\hat{\gamma}^+}}}\right). \tag{7}$$

Thus, we reach our conclusion.

Existance of $\sum_{k=1}^{K} \hat{\mathbf{B}}_k$ Here, we provides more details about the analysis of existence of $\sum_{k=1}^{K} \hat{\mathbf{B}}_k$, i.e., $\exists\{\hat{\mathbf{B}}_k\}_{k=1}^{K}, s.t.,\ \hat{R}_{\mathcal{D}^{tr}_{\mathcal{T}_t}}(\theta_{0t}) \leq \hat{R}_{\mathcal{D}^{tr}_{\mathcal{T}_t}}(\theta_0)$. Though the negative gradient descent, we can get

$$\begin{aligned}\hat{\theta}_0 &= \theta_0 - \alpha\nabla\mathcal{L}_{\theta,} \\ &= (\mathbf{I} - \alpha\nabla\mathcal{L}(\theta_0)(\theta_0\mathbf{I})^{-1})\theta_0.\end{aligned} \tag{8}$$

Then, we can find a $\sum_{k=1}^{K}\hat{\mathbf{B}}_k = \mathbf{I} - \alpha\nabla\mathcal{L}(\theta_0)(\theta_0\mathbf{I})^{-1}$. It can also be verified in Figure 1. Assume θ_0 is in the red contour, we can find a better parameter $\hat{\theta}_0$ inside the contour through its negative gradient direction.

B. Detailed Description of the New Few-shot Classification Benchmark

The new benchmark consists of four image classification datasets. All images are resized to $84 \times 84 \times 3$. Here, we briefly introduce each of them as follows:

- **Caltech-UCSD Birds-200-2011 (CUB-200-2011)** (Wah et al., 2011) is a bird image dataset which contains 11,788 photos of 200 bird species. In this paper, we randomly select 100 species with 60 photos in each species. We split the meta-training/meta-validation/meta-testing sets as 64/16/20 species.

- **Meta-training:** Savannah Sparrow, Dark eyed Junco, Black footed Albatross, Henslow Sparrow, Cape Glossy Starling, Black throated Sparrow, Northern Waterthrush, Hooded Warbler, Baltimore Oriole, Scarlet Tanager, Cerulean Warbler, Downy Woodpecker, Black and white Warbler, Tropical Kingbird, Canada Warbler, Blue Jay, Elegant Tern, Groove billed Ani, Mallard, European Goldfinch, Red breasted Merganser, Geococcyx, Red winged Blackbird, Ringed Kingfisher, Prairie Warbler, Florida Jay, Hooded Oriole, American Redstart, Western Wood Pewee, Sayornis, Myrtle Warbler, Yellow Warbler, Tree Swallow, Rufous Hummingbird, Fish Crow, Bewick Wren, Seaside Sparrow, Vesper Sparrow, American Crow, Eared Grebe, Blue headed Vireo, White necked Raven, Frigatebird, Horned Lark, Tree Sparrow, Red bellied Woodpecker, Pacific Loon, Caspian Tern, Anna Hummingbird, Olive sided Flycatcher, Common Tern, Cedar Waxwing, Great Crested Flycatcher, Blue Grosbeak, White breasted Kingfisher, White eyed Vireo, Purple Finch, Cliff Swallow, Scissor tailed Flycatcher, Harris Sparrow, Western Grebe, Gadwall, American Goldfinch, Pine Warbler.
- **Meta-validation:** Mockingbird, Vermilion Flycatcher, Cape May Warbler, Prothonotary Warbler, White crowned Sparrow, Ovenbird, Pomarine Jaeger, Indigo Bunting, Blue winged Warbler, Chipping Sparrow, Horned Grebe, Fox Sparrow, Green Violetear, Nashville Warbler, Least Tern, Marsh Wren.
- **Meta-testing:** Rose breasted Grosbeak, Nighthawk, Long tailed Jaeger, Bronzed Cowbird, California Gull, Ivory Gull, Northern Fulmar, Brown Pelican, Ring billed Gull, Great Grey Shrike, White breasted Nuthatch, Mourning Warbler, Sage Thrasher, Horned Puffin, Pied Kingfisher, Shiny Cowbird, Scott Oriole, Red eyed Vireo, Song Sparrow, Winter Wren.

- **Describable Textures Dataset (DTD)** (Cimpoi et al., 2014) is a texture image dataset which contains 5640 images from 47 classes. Each class contains 120 images. Meta-training/Meta-validation/Meta-testing contains 30/7/10 classes respectively.
  - **Meta-training:** pitted, woven, crosshatched, crystalline, sprinkled, lacelike, bubbly, marbled, dotted, bumpy, striped, zigzagged, lined, smeared, pleated, stratified, waffled, knitted, gauzy, porous, spiralled, grooved, banded, potholed, stained, veined, swirly, frilly, freckled, studded.
  - **Meta-validation:** wrinkled, grid, perforated, cobwebbed, honeycombed, cracked, blotchy.
  - **Meta-testing:** fibrous, matted, scaly, chequered, flecked, paisley, braided, polka-dotted, interlaced, meshed.

- **Fine-Grained Visual Classification of Aircraft (FGVC-Aircraft)** (Maji et al., 2013) is a image dataset for fine grained visual categorization of aircraft. The dataset contains 102 different aircraft variants. In this paper, we randomly select 100 variants with 100 images in each variant. We split the meta-training/meta-validation/meta-testing to 64/16/20 variants respectively.
  - **Meta-training:** MD-90, 737-600, A310, An-12, DR-400, Falcon-900, DC-3, Challenger-600, Fokker-70, Cessna-172, 747-400, ERJ-145, Dornier-328, A330-300, A319, Model-B200, E-170, A340-500, BAE-125, Metroliner, 747-300, C-130, DH-82, Hawk-T1, 727-200, 767-300, DC-10, Spitfire, E-195, BAE-146-300, F-16A-B, Beechcraft-1900, 747-200, Boeing-717, Falcon-2000, 777-300, Cessna-560, DHC-8-100, Cessna-525, 737-200, DC-8, Global-Express, DHC-1, CRJ-200, A340-300, DC-9-30, CRJ-900, A320, 737-300, Eurofighter-Typhoon, SR-20, E-190, Saab-340, C-47, Il-76, MD-87, 757-300, DHC-6, Tu-154, 777-200, 767-200, A318, 757-200, A300B4.
  - **Meta-validation:** 737-900, A340-600, 737-800, 737-400, L-1011, A330-200, Gulfstream-V, 737-500, A340-200, ATR-72, MD-11, CRJ-700, EMB-120, Fokker-100, DC-6, 737-700.
  - **Meta-testing:** 707-320, PA-28, Cessna-208, F-A-18, DHC-8-300, ERJ-135, Tornado, BAE-146-200, A321, ATR-42, Saab-2000, Tu-134, Fokker-50, A380, MD-80, Gulfstream-IV, Yak-42, 747-100, 767-400, Embraer-Legacy-600.

- **FGVCx-Fungi (Fungi)** (Fun, 2018) contains over 100,000 fungi images of nearly 1,500 wild mushroom species. We first filter the species with less than 150 images and then randomly select 100 species with 150 images in each species. We split the meta-training/meta-validation/meta-testing to 64/16/20 species respectively.
  - **Meta-training**: Suillus granulatus, Phaeolus schweinitzii, Cystoderma amianthinum, Pycnoporellus fulgens, Psathyrella candolleana, Meripilus giganteus, Phellinus pomaceus, Laccaria laccata, Laccaria proxima, Amanita excelsa, Ganoderma pfeifferi, Clitopilus prunulus, Agaricus arvensis, Hericium coralloides, Plicatura crispa, Agrocybe praecox, Steccherinum ochraceum, Hypholoma fasciculare, Xerocomellus pruinatus, Xerocomellus chrysenteron, Crepidotus cesatii, Auricularia auricula-judae, Heterobasidion annosum, Entoloma clypeatum, Cortinarius torvus, Mycena tintinnabulum, Laetiporus sulphureus, Datronia mollis, Pholiota squarrosa, Cerioporus squamosus, Tricholoma terreum, Coprinellus micaceus, Cylindrobasidium laeve, Dacrymyces stillatus, Gloeophyllum sepiarium, Lycoperdon perlatum, Hygrophorus pustulatus, Clavulina coralloides, Xerocomus ferrugineus, Cortinarius alboviolaceus, Byssomerulius corium, Boletus edulis, Hymenopellis radicata, Basidioradulum radula, Cortinarius elatior, Schizophyllum commune, Cortinarius malicorius, Suillellus luridus, Ganoderma applanatum, Oligoporus guttulatus, Tubaria furfuracea, Cortinarius largus, Pleurotus ostreatus, Stereum hirsutum, Xylodon raduloides, Peniophora incarnata, Sutorius luridiformis, Flammulina velutipes var. velutipes, Phlebia radiata, Hygrocybe conica, Chlorophyllum olivieri, Armillaria ostoyae, Peniophora quercina, Mycena galericulata
  - **Meta-validation:** Agaricus impudicus, Daedaleopsis confragosa, Fomitopsis pinicola, Cortinarius anserinus, Mucidula mucida, Trametes versicolor, Stropharia cyanea, Ramaria stricta, Radulomyces confluens, Gliophorus psittacinus, Psathyrella spadiceogrisea, Coprinopsis lagopus, Daedalea quercina, Amanita muscaria, Armillaria lutea, Vuilleminia comedens

– **Meta-testing:** Hygrocybe ceracea, Trametes hirsuta, Polyporus tuberaster, Lacrymaria lacrymabunda, Fistulina hepatica, Gymnopus dryophilus, Amanita rubescens, Fuscoporia ferrea, Craterellus undulatus, Tricholoma scalpturatum, Mycena pura, Russula depallens, Bjerkandera adusta, Trametes gibbosa, Tremella mesenterica, Cerioporus varius, Amanita fulva, Xylodon paradoxus, Cuphophyllus virgineus, Cortinarius flexipes

C. Hyperparameters & Additional Experiment Settings

We summarize the hyperparameters in this paper in Table 1. Like (Finn et al., 2017), we compute the full Hessian-vector products for MAML. All cluster centers are randomly initialized. Note that, in few-shot classification problem, we use the change of averaged training accuracy to determine whether to increase clusters. Thus, $\mu < 1$ in this problem. For toy regression task, the pre-aggregator embedding $\mathcal{F}(\cdot, \cdot)$ is a fully connected layer. Following (Finn et al., 2017), the base learner has two hidden layers with 40 neurons in each. For few-shot image classification task, the pre-aggregator embedding $\mathcal{F}(\cdot, \cdot)$ is a block of two convolutional layers with two fully connected layers. The base learner is a standard base learner with 4 standard convolutional blocks. For continual scenario, we add one cluster every time. All the experiments are implemented using Tensorflow (Abadi et al., 2016).

Table 1. Hyperparameter summary

Hyperparameters	Toy Regreesion	miniImageNet	Multi-Datasets (New Benchmark)
Input Scale (only for image data)	/	$84 \times 84 \times 3$	$84 \times 84 \times 3$
Meta-batch Size (task batch size)	25	4	4
Inner loop learning rate (α)	0.001	0.001	0.001
Outer loop learning rate (β)	0.001	0.01	0.01
Filters of CNN (only for image data)	/	32	32
Meta-training adaptation steps	5	5	5
Task representation size	40	128	128
Reconstruction loss weight (γ)	0.01	0.01	0.01
Image Embedding Size (before aggregator)	/	64	64
Continual Training Threshold (τ)	1.25	/	0.85
# epoch (Q) for computing loss	1000	/	100

D. Results of MiniImagenet

In this part, we present the additional comparison on MiniImagenet dataset. Similar to the analysis in (Finn et al., 2018), the sampled tasks in this benchmark do not have obvious heterogeneity and uncertainty. Thus, the goal is to compare our approach with gradient-based meta-learning methods and other previous models. The expressive capacity of each model is controlled by using 4 standard convolutional layers and the results are shown in Table 2. With the same expressive capacity, our model can achieve comparable performance with MAML-based models and other previous models in meta-learning field.

E. Leave-one-out Experiments on Few-shot Image Classification

In this part, we design a more difficult experiment for few-shot image classification. For each dataset, we use three datasets for meta-training and the remaining dataset for meta-testing. For example, we use texture, bird and aircraft datasets for meta-training, and fungi dataset for meta-testing. Different from all the previous meta-learning settings which only use different classes for meta-testing, the leave-one-out experiment use a totally different dataset to test the generalization performance, which is more challenging.

The results of 5-way 1-shot classification are shown in Table 3. We compare our methods with MAML and MUMOMAML (the best baseline in few-shot classification). We can see all results are significantly worse than the results without the leave-one-out technique, which shows the difficulty of this experiment. However, by capturing task clustering structure, our method can still achieves better performance than MAML and MUMOMAML.

Table 2. Comparison between our approach and prior few-shot learning techniques on the 5-way, 1-shot MiniImagenet benchmark. For MT-Net (Lee & Choi, 2018), we remove the T-block since it introduces several 1×1 convolutional layers which increases the expressive capacity of base learner (Lin et al., 2013). For BMAML (Yoon et al., 2018), 24 classes are used for meta-testing in their original paper, while other methods use 20 classes. Since they have not released their code, we are not able to know the used classes. Thus, we implement it and report their performance on the standard classes (i.e., 20 classes for testing). Like (Finn et al., 2018), we bold methods whose highest scores that overlap in their confidence intervals.

MiniImagenet	5-way 1-shot Accuracy
Matching Nets (Vinyals et al., 2016)	$43.56 \pm 0.34\%$
meta-learner LSTM (Ravi & Larochelle, 2016)	$43.44 \pm 0.77\%$
Prototypical Network (Snell et al., 2017)	$46.61 \pm 0.78\%$
SNAIL (Mishra et al., 2018)	$45.10 \pm 0.00\%$
mAP-DLM (Triantafillou et al., 2017)	$49.82 \pm 0.78\%$
Relation Net (Yang et al., 2018)	$\mathbf{50.44 \pm 0.82\%}$
GNN (Garcia & Bruna, 2017)	$\mathbf{50.33 \pm 0.36\%}$
MAML (Finn et al., 2017)	$48.70 \pm 1.84\%$
LLAMA (Finn & Levine, 2017)	$49.40 \pm 1.83\%$
BMAML (Yoon et al., 2018)	$50.01 \pm 1.86\%$
MT-Net (Lee & Choi, 2018)	$49.75 \pm 1.83\%$
MUMOMAML (Vuorio et al., 2018)	$49.86 \pm 1.85\%$
Reptile (Nichol & Schulman, 2018)	$49.97 \pm 0.32\%$
MetaSGD (Li et al., 2017)	$\mathbf{50.47 \pm 1.87\%}$
PLATIPUS (Finn et al., 2018)	$\mathbf{50.13 \pm 1.86\%}$
HSML (ours)	$\mathbf{50.38 \pm 1.85\%}$

Table 3. Comparison of leave-one-out experiments on 5-way 1-shot classification. 4000 tasks are used to test the performance. For each dataset, the performance is reported when this dataset is used for meta-testing.

Model	Bird	Texture	Aircraft	Fungi	Average
MAML	$40.76 \pm 0.68\%$	$29.50 \pm 0.65\%$	$29.54 \pm 0.63\%$	$29.94 \pm 0.64\%$	32.43%
MUMOMAML	$41.58 \pm 0.68\%$	$30.24 \pm 0.68\%$	$30.69 \pm 0.66\%$	$30.63 \pm 0.66\%$	33.28%
HSML-RTG	$\mathbf{42.54 \pm 0.67\%}$	$\mathbf{30.90 \pm 0.67\%}$	$\mathbf{31.23 \pm 0.64\%}$	$\mathbf{32.98 \pm 0.68\%}$	$\mathbf{34.41\%}$

F. Additional Results of Few-shot Classification

Table 4 and Table 5 contain the full results (accuracy with 95% confident interval) of few-shot image classfiation. Table 4 shows the full results of the bottom table in Figure 7 (in paper). Table 5 contains the full results of Table 3 (in paper).

Table 4. Comparison of online update results on few-shot image classification 5-way 1-shot scenario (Full Table).

Model	Bird	Texture	Aircraft	Fungi	Average
MUMOMAML	$56.66 \pm 1.43\%$	$33.68 \pm 1.37\%$	$45.73 \pm 1.39\%$	$40.38 \pm 1.40\%$	44.11%
HSML-Static (2C)	$60.77 \pm 1.43\%$	$33.41 \pm 1.40\%$	$51.28 \pm 1.37\%$	$40.78 \pm 1.34\%$	46.56%
HSML-Static (10C)	$59.16 \pm 1.49\%$	$34.48 \pm 1.36\%$	$52.30 \pm 1.35\%$	$40.56 \pm 1.39\%$	46.63%
HSML-Dynamic	$\mathbf{61.16 \pm 1.42\%}$	$\mathbf{34.53 \pm 1.35\%}$	$\mathbf{54.50 \pm 1.36\%}$	$\mathbf{41.66 \pm 1.41\%}$	$\mathbf{47.96\%}$

Table 5. Comparison of different cluster numbers (Full Table).

Num. of Clus.	Bird	Texture	Aircraft	Fungi	Average
$(2, 2, 1)$	$58.37 \pm 1.42\%$	$33.18 \pm 1.34\%$	$56.15 \pm 1.36\%$	$42.90 \pm 1.41\%$	47.65%
$(4, 2, 1)$	$\mathbf{60.98 \pm 1.50\%}$	$\mathbf{35.01 \pm 1.36\%}$	$57.38 \pm 1.40\%$	$44.02 \pm 1.39\%$	$\mathbf{49.35\%}$
$(6, 3, 1)$	$60.55 \pm 1.45\%$	$34.02 \pm 1.34\%$	$55.79 \pm 1.38\%$	$43.43 \pm 1.39\%$	48.45%
$(8, 4, 4, 1)$	$59.55 \pm 1.46\%$	$34.74 \pm 1.37\%$	$\mathbf{57.83 \pm 1.39\%}$	$\mathbf{44.18 \pm 1.38\%}$	49.08%

G. Effect of Different Aggregator

In our experiment, we found that the recurrent aggregator performs the best. To give more quantitative insight about the choice of aggregator, we compare these two aggregators with different shots in Table 6. We can see that recurrent aggregator significantly outperforms in 1-shot scenario. With the increase of the size of training samples, the performances of the two aggregators become more similar. Therefore, compared with recurrent aggregator, training a better mean pooling aggregator may require more data.

Table 6. Comparison of different aggregator on different shot, where HSML-RAA and HSML-MPAA represent HSML with recurrent autoencoder aggregator and mean pooling autoencoder aggregator, respectively.

	Model	Bird	Texture	Aircraft	Fungi	Average
1-shot	HSML-MPAA	$57.87 \pm 1.48\%$	$32.07 \pm 1.36\%$	$53.76 \pm 1.41\%$	$40.88 \pm 1.37\%$	46.14%
	HSML-RAA	$\mathbf{60.98 \pm 1.50\%}$	$\mathbf{35.01 \pm 1.36\%}$	$\mathbf{57.38 \pm 1.40\%}$	$\mathbf{44.02 \pm 1.39\%}$	$\mathbf{49.35\%}$
3-shot	HSML-MPAA	$67.80 \pm 0.91\%$	$44.33 \pm 0.82\%$	$67.73 \pm 0.83\%$	$52.45 \pm 0.94\%$	58.07%
	HSML-RAA	$\mathbf{68.01 \pm 0.88\%}$	$\mathbf{45.07 \pm 0.87\%}$	$\mathbf{68.59 \pm 0.82\%}$	$\mathbf{53.51 \pm 0.96\%}$	$\mathbf{58.80\%}$
5-shot	HSML-MPAA	$\mathbf{71.80 \pm 0.70\%}$	$48.02 \pm 0.68\%$	$71.79 \pm 0.74\%$	$54.01 \pm 0.82\%$	61.40%
	HSML-RAA	$71.68 \pm 0.73\%$	$\mathbf{48.08 \pm 0.69\%}$	$\mathbf{73.49 \pm 0.68\%}$	$\mathbf{56.32 \pm 0.80\%}$	$\mathbf{62.39\%}$
8-shot	HSML-MPAA	$\mathbf{75.75 \pm 0.62\%}$	$\mathbf{52.90 \pm 0.57\%}$	$73.03 \pm 0.55\%$	$\mathbf{58.20 \pm 0.73\%}$	64.97%
	HSML-RAA	$75.52 \pm 0.63\%$	$51.52 \pm 0.59\%$	$\mathbf{75.33 \pm 0.53\%}$	$57.68 \pm 0.71\%$	$\mathbf{65.01\%}$

H. Ablation Studies

To investigate the contribution of different components of HSML (i.e., task representation, hierarchical task clustering, knowledge adaptation), we conduct the following ablation studies from four perspectives in Table 7, where 5-way, 1-shot results on image classification are reported. The detailed ablations are provided in follows:

- (A1) We train four MAMLs for four clusters, i.e., bird, texture, aircraft and fungi, by assigning a task to its groundtruth cluster. The results can be regarded as an upper-bound application of MAML with task clustering, provided with groundtruth clusters of all tasks which are unfortunately absent in real-world applications. HSML outperforms as the soft and hierarchical clustering not only accurately captures the task relationship but also encourages knowledge transfer across clusters.
- (A2) We investigate different variants of task representation learning in (A2a) and (A2b). In (A2a), we first use reconstruction loss to train task embeddings. Next, we fix the parameters of the task representation learning component and backpropagate meta-gradients to only train the other two components. The results are inferior, showing that meta-learning gradients further optimize task embeddings. In (A2b), we replace our task embedding with the last hidden state of the encoder. The results higher than MUMOMAML show the contribution of hierarchical clustering, while they worse than ours further justify the capability of our task embedding.
- (A3) We analyze the effect of hierarchical clustering in (A3). In (A3a), we remove the hierarchical task clustering component. In (A3b), we consider the flat instead of hierarchical task clustering. The results of (A3a) lower than (A3b) consolidate our motivation of knowledge generalization with a cluster.
- (A4) We also study three variants of knowledge adaptation in (A4a)-(A4c). In (A4a), we revise Eqn. (8) by only using the clustering representation. The results still compete with state-of-the-art baselines, but empirically the combination with the task representation yields the best performance. In (A4b), we replace the parameter gate $\mathbf{o}_i$ with FiLM (Perez et al., 2018), where the comparable results verify the primary contribution of hierarchical clustering. In order to validate the effectiveness of parameter gate, in (A4c), we directly learn the initialization from the task representation and the cluster representation instead of using the parameter gate to mask a shared initialization. The poor results show that the parameter gate masking a shared set of parameters θ_0 may 1) prevent the curse of dimensionality and constrain the optimization space, given the high dimensionality of parameters; 2) serve as the warm-start for a new cluster of tasks in continual learning.

Table 7. Ablation Studies. Results of 5-way, 1-shot image classification are reported.

Ablation	Bird	Texture	Aircraft	Fungi
(**A1**): Train a MAML for each cluster, e.g., bird, by assigning a task to its groundtruth cluster.	$58.25 \pm 1.46\%$	$34.53 \pm 1.36\%$	$55.73 \pm 1.37\%$	$43.59 \pm 1.39\%$
(**A2a**): Use reconstruction loss to pretrain the task representation learning component, and then fix the paramters of it and backpropagate meta-gradients to only train the other two components.	$56.97 \pm 1.44\%$	$29.12 \pm 1.30\%$	$45.71 \pm 1.38\%$	$40.92 \pm 1.39\%$
(**A2b**): Replace our task embedding with the last hidden state of the encoder.	$58.25 \pm 1.49\%$	$34.53 \pm 1.36\%$	$55.73 \pm 1.37\%$	$43.59 \pm 1.39\%$
(**A3a**): Remove the hierarchical task clustering component.	$58.22 \pm 1.48\%$	$33.30 \pm 1.36\%$	$55.35 \pm 1.38\%$	$42.68 \pm 1.40\%$
(**A3b**): Consider only flat rather than hierarchical task clustering.	$58.08 \pm 1.45\%$	$34.26 \pm 1.35\%$	$56.11 \pm 1.38\%$	$43.38 \pm 1.39\%$
(**A4a**): Infer the parameter gate with the clustering representation only.	$59.01 \pm 1.50\%$	$33.69 \pm 1.35\%$	$56.69 \pm 1.39\%$	$42.88 \pm 1.40\%$
(**A4b**): Replace the parameter gate with FiLM (Perez et al., 2018).	$61.02 \pm 1.47\%$	$34.87 \pm 1.37\%$	$56.53 \pm 1.40\%$	$44.56 \pm 1.38\%$
(**A4c**): Learn the initialization directly from task and cluster representations rather than using the parameter gate.	$53.95 \pm 1.47\%$	$32.35 \pm 1.35\%$	$52.15 \pm 1.37\%$	$42.31 \pm 1.40\%$

I. Additional Task Clustering Results of Toy Regression Tasks

In Figure I, we show the additional results of task clustering analysis of toy regression. In this figure, we further verify that tasks can be clustered by their shapes. Clusters 1 reflects the fluctuation mode curve (e.g., Sin a1-a4, Cubic a1-a4), while cluster 2 reflects an arc (e.g., Quad a2-a4). Cluster 3 mainly reflects a linear shape with positive slope (e.g. Line a1, Line a2, Quad a1, Quad a2, Cubic a1). Cluster 4 mainly reflects a linear shape with negative slope (e.g., Line a3, Line a4, Cubic a4).

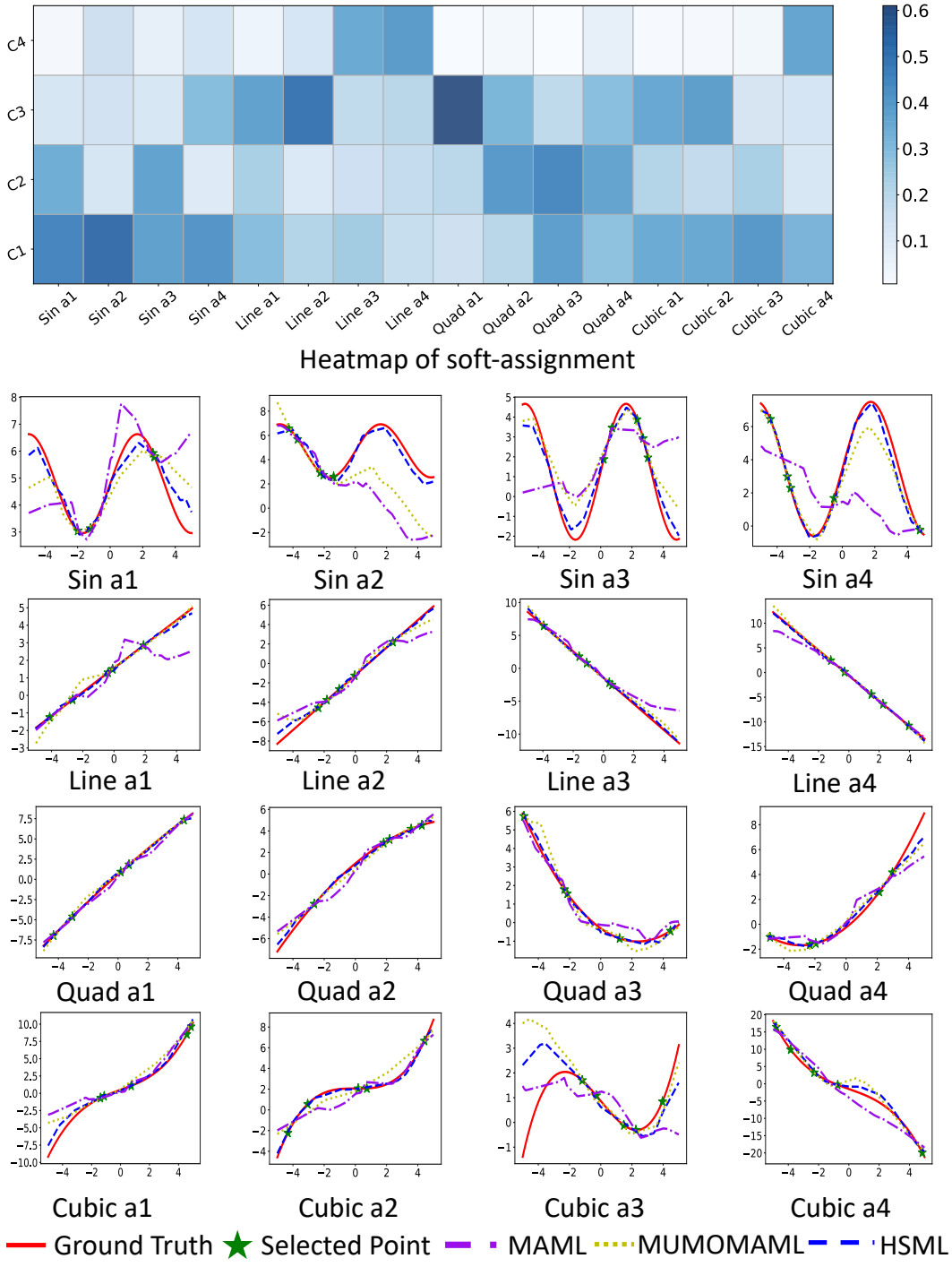


Figure 2. Additional results of task clustering analysis of toy regression problem.

J. Additional Task Clustering Analysis of Few-shot Classification

In Figure 3, we show the additional results of task clustering analysis. The soft-assignment heatmap with their training images and activation paths of twelve tasks are illustrated. The conclusion is similar to that we draw previously in the paper. Tasks from different datasets mainly activate different clusters: bird→cluster 2, texture→cluster 4, aircraft→cluster 1, fungi→cluster 3. The left cluster and right cluster in the second layer may represent environment and surface texture, respectively.

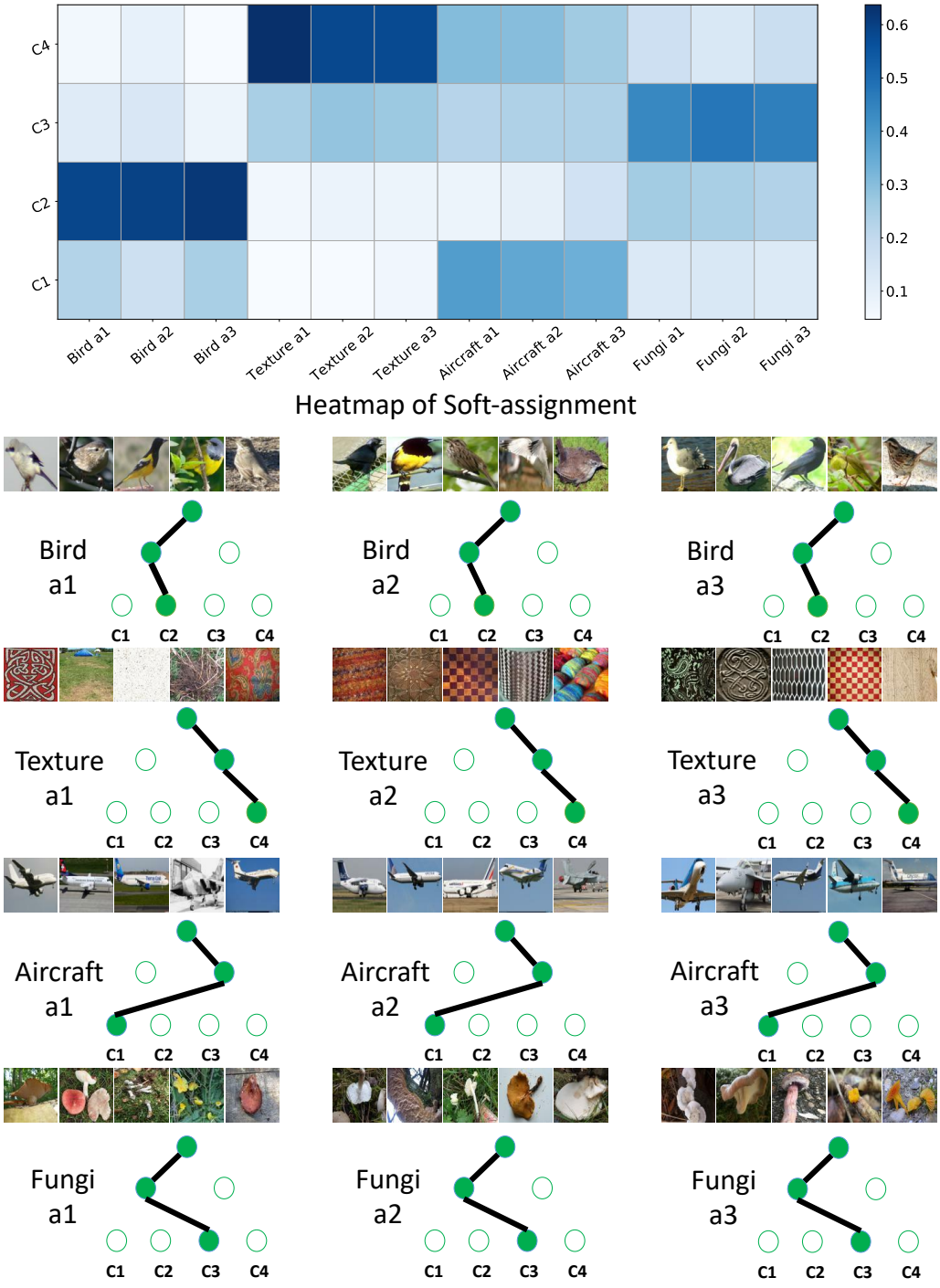


Figure 3. Additional results of task clustering analysis of few-shot image classification problem.

References

2018 fgcvx fungi classification challenge, 2018. URL https://www.kaggle.com/c/fungi-challenge-fgvc-2018.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , and Vedaldi, A. Describing textures in the wild. In *CVPR*, 2014.

Finn, C. and Levine, S. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *arXiv preprint arXiv:1710.11622*, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pp. 1126–1135, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Garcia, V. and Bruna, J. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

Kuzborskij, I. and Lampert, C. H. Data-dependent stability of stochastic gradient descent. *arXiv preprint arXiv:1703.01678*, 2017.

Lee, Y. and Choi, S. Gradient-based meta-learning with learned layerwise metric and subspace. In *ICML*, pp. 2933–2942, 2018.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Lin, M., Chen, Q., and Yan, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Maji, S., Kannala, J., Rahtu, E., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. Technical report, 2013.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. *ICLR*, 2018.

Nichol, A. and Schulman, J. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. C. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. *ICLR*, 2016.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *NIPS*, pp. 4077–4087, 2017.

Triantafillou, E., Zemel, R., and Urtasun, R. Few-shot learning through an information retrieval lens. In *NIPS*, pp. 2255–2265, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *NIPS*, pp. 3630–3638, 2016.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. Toward multimodal model-agnostic meta-learning. *arXiv preprint arXiv:1812.07172*, 2018.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Yang, F. S. Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *NIPS*, pp. 7343–7353, 2018.