HyperShot: Few-Shot Learning by Kernel HyperNetworks – supplementary material

Marcin Sendera [†1] Marcin Przewięźlikowski[†1] Konrad Karanowski[2]
Maciej Zięba[2,3] Jacek Tabor[1] Przemysław Spurek[1]

[1]Faculty of Mathematics and Computer Science, Jagiellonian University
6 Łojasiewicza Street, 30-348 Kraków, Poland

[2]Department of Artificial Intelligence, University of Science and Technology
Wyb. Wyspiańskiego 27, 50-370, Wrocław, Poland

[3]Tooploox
Tęczowa 7, 53-601, Wrocław, Poland
marcin.{sendera, przewiezlikowski}@doctoral.uj.edu.pl

A. Additional results

This section provides additional results in the 5-way (1-shot and 5-shot) classification tasks for models using larger backbones, namely ResNet-10 and ResNet-12 [11], as well as and expanded version of Table 1 from the main text, containing more baselines . We provide the results for ResNet-10 on the **CUB** and **mini-ImageNet** datasets in Table 1, for ResNet-12 on **mini-ImageNet** dataset in Table 2 and for Conv4 on the **CUB** and **mini-ImageNet** datasets in Table 3. It should be noted that the results for ResNet-10 on **mini-ImageNet** for all methods were obtained by us using a unified codebase [3, 27]. On the other hand, for benchmarks of ResNet-10 on **CUB** and ResNet-12 on **mini-ImageNet** we report the accuracies of methods other than HyperShot as reported in [27] and [45], respectively.

ResNet-10 – CUB and mini-ImageNet In the **CUB** dataset classification tasks (see Table 1), HyperShot is amongst the state-of-the-art models achieving classification accuracy often equal within the variance to the best models. Considering the 5-shot scenario, the highest classification result across the evaluated methods ($86.38\% \pm 0.15$) obtained the GPLDLA model based on the Gaussian Processes framework. However, the HyperShot performance, $86.28\% \pm 0.29$, is the second-best but even lies within the variance of the best model. In the 1-shot setting, ProtoNet obtains the highest result ($73.22\% \pm 0.92$), whereas HyperShot is the third one ($71.99\% \pm 0.70$) but still equal according to the variances.

In the **mini-ImageNet** classification task with the ResNet-10 backbone, HyperShot achieves the second-best accuracy in both 1-shot and 5-shot settings[*]. In the 1-shot setting, the DKT model [27] achieved the best result, with HyperShot being a close second, with only 0.04 pp difference. In the 5-shot setting, the baseline++ approach outperforms all others by a large margin [3], whereas HyperShot and ProtoNet [35] achieve similar, second-best results. We observe that apart from HyperShot, which achieves second-best results in both settings, models which perform well in one setting are outperformed by others in the second and vice versa.

ResNet-12 – mini-ImageNet In the **mini-ImageNet** classification task with the ResNet-12 backbone (see Table 2), HyperShot ranks relatively low in terms of accuracy (63.30% and 76.21% in the 1-shot and 5-shot settings, respectively), as compared to recently proposed approaches such as RENet [14] and FEAT [45], which outperform it by a large margin. Nevertheless, we note that for the experiments on ResNet-12 we used the same set of hyperparameters as in the ResNet-10 experiments, which may not be an optimal choice for a larger backbone.

[†]Denotes equal contribution.

[*]In the case of the **mini-ImageNet** classification with ResNet10, we benchmarked all of the listed models ourselves. To our best knowledge, previously, there were no reported benchmarks on this dataset with the ResNet-10 backbone.

It is worth noticing that HyperShot without adaptation steps performances sometimes slightly better than the same with adaptation. We even observe that a few first steps of adaptation procedure result in an unnoticeable increase of accuracy of the basic model. However, the usual 10 steps result in this setting in slightly worse performance, so one should use it cautiously. We decided to report the results after the standard adaptation procedure only.

B. Training details

In this section, we present in detail the architecture and hyperparameters of HyperShot.

Architecture overview From a high-level perspective, the architecture of HyperShot consists of three parts:

- backbone - a convolutional feature extractor.
- neck - a sequence of zero or more fully-connected layers with ReLU nonlinearities in between.
- heads - for each parameter of the target network, a sequence of one or more linear layers, which predicts the values of that parameter. All heads of HyperShot have identical lengths, hidden sizes, and input sizes that depend on the generated parameter's size.

The target network generated by HyperShot re-uses its backbone. We outline this architecture in Figure 1.

Backbone For each experiment described in the main body of this work, we follow [27] in using a shallow backbone (feature extractor) for HyperShot as well as referential models. This backbone consists of four convolutional layers, each consisting of a convolution, batch normalization, and ReLU nonlinearity. Apart from the first convolution, which has the number of input size equal to the number of image channels, each convolution has an input and output size of 64. We apply max-pooling between each convolution, which decreases by half the resolution of the processed feature maps. The output of the backbone is flattened so that the further layers can process it.

We perform additional experiments described in Appendix A where instead of the above backbone, we utilize ResNet-10 [11].

Datasets For the purpose of making a fair comparison, we follow the procedure presented in, e.g., [27, 3]. In the case of the **CUB** dataset [41], we split the whole amount of 200 classes (11788 images) across train, validation, and test consisting of 100, 50, and 50 classes, respectively [3]. The **mini-ImageNet** dataset [32] is created as the subset of **ImageNet** [34], which consists of 100 different classes represented by 600 images for each one. We followed the standard procedure and divided the **mini-ImageNet** into 64 classes for the train, 16 for the validation set, and the remaining 20 classes for the test. The well-known **Omniglot** dataset [17] is a collection of characters from 50 different languages. The **Omniglot** contains 1623 white and black characters in total. We utilize the standard procedure to include the examples rotated by 90° and increase the size of the dataset to 6492, from which 4114 were further used in training. Finally, the **EMNIST** dataset [4] collects the characters and digits coming from the English alphabet, which we split into 31 classes for the test and 31 for validation.

Data augmentation We apply data augmentation during model training in all experiments, except **Omniglot** $\rightarrow$ **EMNIST** cross-domain classification. The augmentation pipeline is identical to the one used by [27] and consists of the random crop, horizontal flip, and color jitter steps.

C. Hyperparameters

Below, we outline the hyperparameters of architecture and training procedures used in each experiment.

We use cosine similarity as a kernel function and averaged support embeddings aggregation in all experiments. HyperShot is trained with the learning rate of 0.001 with the Adam optimizer [16] and no learning rate scheduler. Task-specific adaptation is also performed with the Adam optimizer and the learning rate of 0.0001.

For the natural image tasks (**CUB**, **mini-ImageNet**, **mini-ImageNet** $\rightarrow$ **CUB** classification), we use a hypernetwork with the neck length o 2, head lengths of 3, and a hidden size of 4096, which produce a target network with a single fully-connected layer. We perform training for 10000 epochs.

For the simpler **Omniglot** $\rightarrow$ **EMNIST** character classification task, we train a smaller hypernetwork with the neck length of 1, head lengths of 2, and the hidden size of 512, which produces a target network with two fully-connected layers and a hidden size of 128. We train this hypernetwork for a shorter number of epochs, namely 2000.

We summarize all the above hyperparameters in Table 4.

D. Source code

The source code required for running the experiments is available at `https://github.com/gmum/few-shot-hypernets-public`.

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml, 2018.

[2] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form

Table 1. The classification accuracy results for the inference tasks in the **CUB** and **mini-ImageNet** dataset in the 5-way (1-shot and 5-shot) scenarios. We consider models using the ResNet-10 backbone. The highest results are bold and second-highest in italic (the larger, the better).

	CUB		**mini-ImageNet**	
Method	1-shot	5-shot	1-shot	5-shot
Feature Transfer	63.64 ± 0.91	81.27 ± 0.57	–	–
Baseline++ [3]	69.55 ± 0.89	85.17 ± 0.50	54.35 ± 0.34	$\mathbf{75.26 \pm 0.16}$
MatchingNet [40]	71.29 ± 0.87	83.47 ± 0.58	54.18 ± 0.09	67.71 ± 0.20
ProtoNet [35]	$\mathbf{73.22 \pm 0.92}$	85.01 ± 0.52	53.28 ± 0.17	73.04 ± 0.15
MAML [6]	70.32 ± 0.99	80.93 ± 0.71	–	–
RelationNet [38]	70.47 ± 0.99	83.70 ± 0.55	51.88 ± 0.45	67.21 ± 0.16
DKT + CosSim [27]	70.81 ± 0.52	83.26 ± 0.50	–	–
DKT + BNCosSim [27]	$\mathit{72.27 \pm 0.30}$	85.64 ± 0.29	$\mathbf{56.03 \pm 0.50}$	71.28 ± 0.12
SimpleShot [42]	53.78 ± 0.21	71.41 ± 0.17	–	–
GPLDLA [15]	71.30 ± 0.16	$\mathbf{86.38 \pm 0.15}$	–	–
HyperShot	71.99 ± 0.70	86.28 ± 0.29	55.36 ± 0.64	$\mathit{73.06 \pm 0.30}$
HyperShot + adaptation	71.60 ± 0.59	$\mathit{86.22 \pm 0.30}$	$\mathit{55.99 \pm 0.63}$	72.87 ± 0.33

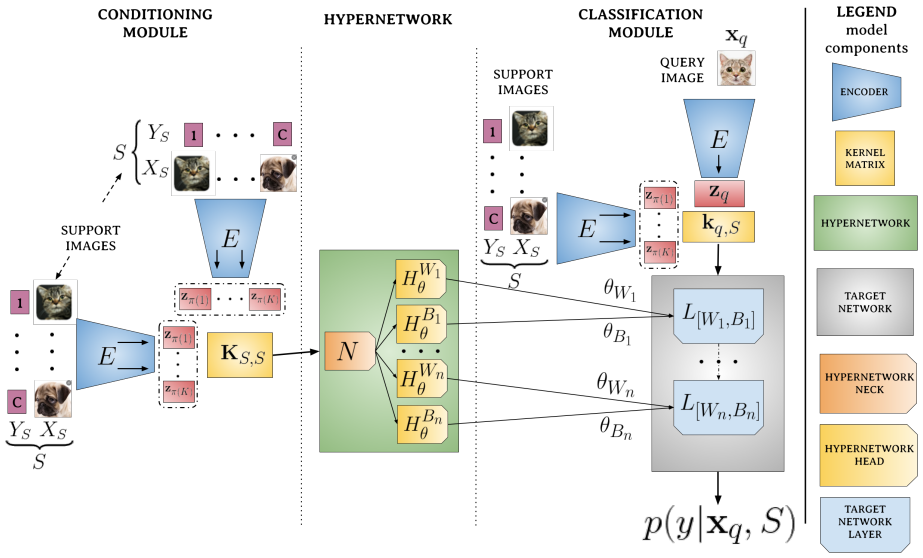

Figure 1. A detailed outline of the architecture of HyperShot, with the denoted flow of parameters generated by the hypernetwork heads.

solvers. In *International Conference on Learning Representations*, 2018.

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[4] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters (2017). *arXiv preprint arXiv:1702.05373*, 2017.

[5] Chen Fan, Parikshit Ram, and Sijia Liu. Sign-maml: Efficient model-agnostic meta-learning by signsgd. *CoRR*, abs/2109.07497, 2021.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[7] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9537–9548, 2018.

[8] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting, 2018.

[9] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2018.

[10] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.

Table 2. The classification accuracy results for the inference tasks in the **mini-ImageNet** dataset in the 5-way (1-shot and 5-shot) scenarios. We consider models using the ResNet-12 backbone. The highest results are bold and second-highest in italic (the larger, the better).

Method	1-shot	5-shot
cosine classifier [3]	55.43 ± 0.81	77.18 ± 0.61
TADAM [26]	58.50 ± 0.30	76.70 ± 0.30
Shot-Free [33]	59.04	77.64
TPN [21]	59.46	75.65
MTL [37]	61.20 ± 1.80	75.50 ± 0.80
RFS-simple [39]	62.02 ± 0.63	79.64 ± 0.44
ProtoNet [35]	62.39 ± 0.21	80.53 ± 0.14
MetaOptNet [18]	62.64 ± 0.82	78.63 ± 0.46
MatchingNet [40]	63.08 ± 0.80	75.99 ± 0.60
MAML [6]	63.11 ± 0.92	–
PPA [28]	59.60	73.34
CAN [12]	63.85 ± 0.48	79.44 ± 0.34
NegMargin [20]	63.85 ± 0.81	81.57 ± 0.56
DeepEMD [47]	65.91 ± 0.82	*82.41 ± 0.56*
FEAT [45]	*66.78 ± 0.20*	82.05 ± 0.14
RENet [14]	**67.60 ± 0.44**	**82.58 ± 0.30**
HyperShot	59.12 ± 0.26	76.53 ± 0.22
HyperShot + adaptation	60.30 ± 0.31	76.21 ± 0.20

Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[12] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification, 2019.

[13] Ghassen Jerfel, Erin Grant, Thomas L Griffiths, and Katherine Heller. Reconciling meta-learning and continual learning with online mixtures of tasks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 9122–9133, 2019.

[14] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification, 2021.

[15] Minyoung Kim and Timothy Hospedales. Gaussian process meta few-shot classifier learning via linear discriminant laplace approximation. *arXiv preprint arXiv:2111.05392*, 2021.

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[17] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[18] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[19] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning, 2017.

[20] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification, 2020.

[21] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning, 2019.

[22] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

[23] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[24] Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Uncertainty in model-agnostic meta-learning using variational inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3090–3100, 2020.

[25] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[26] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

[27] Massimiliano Patacchiola, Jack Turner, Elliot J Crowley, Michael O'Boyle, and Amos J Storkey. Bayesian meta-learning for the few-shot setting via deep kernels. *Advances in Neural Information Processing Systems*, 33, 2020.

[28] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations, 2017.

[29] Jathushan Rajasegaran, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Meta-learning the learning trends shared across tasks. *CoRR*, abs/2010.09291, 2020.

[30] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32:113–124, 2019.

[31] Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2018.

[32] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[33] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training, 2020.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[35] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Table 3. The classification accuracy results for the inference tasks on **CUB** and **mini-ImageNet** datasets in the 1-shot and 5-shot settings. The highest results are in bold and second-highest in italic (the larger, the better). This is an expanded version of Table 1 from the main text.

	CUB		**mini-ImageNet**	
Method	1-shot	5-shot	1-shot	5-shot
ML-LSTM [32]	–	–	43.44 ± 0.77	60.60 ± 0.71
SNAIL [22]	–	–	45.10	55.20
LLAMA [10]	–	–	49.40 ± 1.83	
VERSA [9]	–	–	48.53 ± 1.84	67.37 ± 0.86
Amortized VI [9]	–	–	44.13 ± 1.78	55.68 ± 0.91
Meta-Mixture [13]	–	–	49.60 ± 1.50	64.60 ± 0.92
SimpleShot [42]	–	–	49.69 ± 0.19	66.92 ± 0.17
Feature Transfer [49]	46.19 ± 0.64	68.40 ± 0.79	39.51 ± 0.23	60.51 ± 0.55
Baseline++ [3]	61.75 ± 0.95	78.51 ± 0.59	47.15 ± 0.49	66.18 ± 0.18
MatchingNet [40]	60.19 ± 1.02	75.11 ± 0.35	48.25 ± 0.65	62.71 ± 0.44
ProtoNet [35]	52.52 ± 1.90	75.93 ± 0.46	44.19 ± 1.30	64.07 ± 0.65
RelationNet [38]	62.52 ± 0.34	78.22 ± 0.07	48.76 ± 0.17	64.20 ± 0.28
DKT + CosSim [27]	63.37 ± 0.19	77.73 ± 0.26	48.64 ± 0.45	62.85 ± 0.37
DKT + BNCosSim [27]	62.96 ± 0.62	77.76 ± 0.62	49.73 ± 0.07	64.00 ± 0.09
VAMPIRE [24]	–	–	51.54 ± 0.74	64.31 ± 0.74
PLATIPUS [7]	–	–	50.13 ± 1.86	–
ABML [31]	49.57 ± 0.42	68.94 ± 0.16	45.00 ± 0.60	–
OVE PG GP + Cosine (ML) [36]	63.98 ± 0.43	77.44 ± 0.18	50.02 ± 0.35	64.58 ± 0.31
OVE PG GP + Cosine (PL) [36]	60.11 ± 0.26	79.07 ± 0.05	48.00 ± 0.24	67.14 ± 0.23
Reptile [25]	–	–	49.97 ± 0.32	65.99 ± 0.58
R2-D2 [2]	–	–	48.70 ± 0.60	65.50 ± 0.60
VSM [48]	–	–	54.73 ± 1.60	68.01 ± 0.90
PPA [28]	–	–	54.53 ± 0.40	–
MAML [6]	56.11 ± 0.69	74.84 ± 0.62	45.39 ± 0.49	61.58 ± 0.53
MAML++ [1]	–	–	52.15 ± 0.26	68.32 ± 0.44
iMAML-HF [30]	–	–	49.30 ± 1.88	–
SignMAML [5]	–	–	42.90 ± 1.50	60.70 ± 0.70
Bayesian MAML [46]	55.93 ± 0.71		53.80 ± 1.46	64.23 ± 0.69
Unicorn-MAML [43]	–	–	54.89	–
Meta-SGD [19]	–	–	50.47 ± 1.87	64.03 ± 0.94
MetaNet [23]	–	–	49.21 ± 0.96	–
PAMELA [29]	–	–	53.50 ± 0.89	*70.51 ± 0.67*
FEAT [44]	$\mathbf{68.87 \pm 0.22}$	$\mathbf{82.90 \pm 0.15}$	*55.15 ± 0.20*	$\mathbf{71.61 \pm 0.16}$
DFSVLwF [8]	–	–	$\mathbf{56.20 \pm 0.86}$	–
HyperShot	65.27 ± 0.24	79.80 ± 0.16	52.42 ± 0.46	68.78 ± 0.29
HyperShot+ adaptation	*66.13 ± 0.26*	*80.07 ± 0.22*	53.18 ± 0.45	69.62 ± 0.20

[36] Jake Snell and Richard Zemel. Bayesian few-shot classification with one-vs-each pólya-gamma augmented gaussian processes. In *International Conference on Learning Representations*, 2020.

[37] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning, 2019.

[38] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[39] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need?, 2020.

[40] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Re-

Table 4. Hyperparameters

hyperparameter	**CUB**	**mini-ImageNet**	**mini-ImageNet → CUB**	**Omniglot → EMNIST**
kernel function	cosine similarity	cosine similarity	cosine similarity	cosine similarity
learning rate	0.001	0.001	0.001	0.001
hypernetwork's head layers no.	3	3	3	2
hypernetwork's neck layers no.	2	2	2	1
hypernetwork layers' hidden dim	4096	4096	4096	512
support embeddings aggregation	averaged	averaged	averaged	averaged
taskset size	1	1	1	1
target network layers no.	1	1	1	2
target network activation	ReLU	ReLU	ReLU	ReLU
adaptation epochs (if used)	10	10	10	10
adaptationlearning rate	0.0001	0.0001	0.0001	0.0001
optimizer	Adam	Adam	Adam	Adam
epochs no.	10000	10000	10000	2000

port CNS-TR-2011-001, California Institute of Technology, 2011.

[42] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[43] Han-Jia Ye and Wei-Lun Chao. How to train your maml to excel in few-shot classification, 2021.

[44] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8808–8817, 2020.

[45] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions, 2021.

[46] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7343–7353, 2018.

[47] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Differentiable earth mover's distance for few-shot learning, 2020.

[48] Xiantong Zhen, Ying-Jun Du, Huan Xiong, Qiang Qiu, Cees Snoek, and Ling Shao. Learning to learn variational semantic memory. In *NeurIPS*, 2020.

[49] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2020.