Appendices

A Details on experiments

A.1 Datasets

In our experiments in Section 4, we used the following datasets for meta-training/validation/testing. We used the Torchmeta library [10] for the implementations.

- **Omniglot** Omniglot [29] is a dataset of monochrome 28×28 images of 1623 handwritten characters from 50 different alphabets, which is distributed under the MIT License. For meta-learning, the mutually disjoint 1028/172/423 characters are used for meta-traininig/validation/testing respectively, following Vinyals et al. [59], where each character is randomly rotated by $0, 90, 180, 270$ degrees.
- **CIFAR-FS** CIFAR-FS is a dataset for meta-learning introduced in Bertinetto et al. [8], which consists of 32×32 images with 100 classes from the CIFAR-100 dataset[†][28]. The mutually disjoint 64/16/20 classes [10] are used for meta-training/validation/testing respectively.
- **VGG-Flower** VGG-Flower[†][43] is a dataset of images of 102 species of flowers. For meta-learning, the mutually disjoint 71/16/15 classes [30] are used for meta-training/validation/testing respectively.
- **Aircraft** Aircraft [37] is a dataset of images of 102 classes of aircrafts, which is provided exclusively for non-commercial research purposes. For meta-learning, the mutually disjoint 70/15/15 classes [30] are used for meta-training/validation/testing respectively.
- **miniImageNet** MiniImageNet is a dataset for meta-learning introduced in Vinyals et al [59], which consists of 84×84 images of 100 classes collected from the ImageNet dataset[‡][53]. The mutually disjoint 64/16/20 classes [52, 10] are used for meta-training/validation/testing respectively.
- **CUB** CUB[†] [63] is a dataset of images of 200 species of birds. For meta-learning, the mutually disjoint 100/50/50 classes [10] are used for meta-training/validation/testing respectively.
- **Cars** Cars [27] is a dataset of images of 196 classes of cars, which is provided for research purposes. For meta-learning, the mutually disjoint 98/49/49 classes [57] are used for meta-training/validation/testing respectively.

A.2 Network architectures

A.2.1 5-layered MLPs (for Omniglot in Section 4.1.1)

In Table 3, we summarize the network architecture of 5-layered MLPs used in Section 4.1.1. To analyze the effect of the network size, we introduced the width factor $\rho \in \mathbb{N}$ by which the dimensions of the intermediate outputs are multiplied.

Table 3: The architecture of 5-layered MLPs for Omniglot (ρ: a width factor).

Layers	Output dimensions
Flatten	$784\ (= 28 \times 28)$
Linear $\to$ BatchNorm $\to$ ReLU	256ρ
Linear $\to$ BatchNorm $\to$ ReLU	128ρ
Linear $\to$ BatchNorm $\to$ ReLU	64ρ
Linear $\to$ BatchNorm $\to$ ReLU	64ρ
Linear	# ways (5 or 20)

[†] The licenses of these datasets are unknown.

[‡]ImageNet is provided for non-commercial research or educational use.

A.2.2 5-MLP (for CIFAR-FS, VGG-Flower and Aircraft in Section 4.1.3)

In Table 4, we summarize the network architecture of 5-MLP used in Section 4.1.3. For a fair comparison, the hidden dimensions are chosen so that the baseline method (MAML) achieves a good performance.

Table 4: The architecture of 5-MLP for CIFAR-FS.

Layers	Output dimensions
Flatten	$3072\ (= 3 \times 32 \times 32)$
Linear → BatchNorm → ReLU	1024
Linear → BatchNorm → ReLU	512
Linear → BatchNorm → ReLU	256
Linear → BatchNorm → ReLU	128
Linear	5

A.2.3 CNNs (for miniImageNet, CUB and Cars in Section 4.2)

For ResNet12, we employed the architecture used in Lee et al. [32], following the setting of the BOIL paper by Oh et al. [44]. For WideResNet-28-10, we used the architecture provided in the learn2learn library [4], which is the setting used in Dhillon et al. [11]. The number of parameters for these architectures is summarized in Table 5.

Table 5: The numbers of parameters for CNNs in our experiments.

Networks	# of parameters
ResNet-12	8.0 M
WideResNet-28-10	36.5 M

A.3 Hyperparameters

In our experiments, there are two types of hyperparameters: (1) ones common to gradient-based meta-learning, including MAML and Meta-ticket, and (2) ones specific to Meta-ticket.

A.3.1 Common hyperparameters

Here we summarize hyperparameters common to gradient-based meta-learning: the number of iterations of meta-learning, the number of inner gradient steps S, batch size B for meta-learning, outer learning rate (LR), inner LR, and optimizers. In our experiments, we trained all meta-models (MAML, ANIL, BOIL and Meta-ticket) for 30000 iterations with $S = 1$ and $B = 4$. Other hyperparameters are summarized in Table 6. For MAML-based methods, we followed the settings in the previous work [44].

Table 6: Hyperparameters common to gradient-based meta-learning methods.

Meta-training datasets	Methods	Outer LR	Inner LR	Optimizer
Omniglot	MAML	0.001	0.4	Adam [26]
	Meta-ticket	10.0	0.4	SGD
CIFAR-FS, VGG-Flower	MAML	0.001	0.5	Adam
	Meta-ticket	10.0	0.5	SGD
miniImageNet	MAML	0.0006	0.3	Adam
	Meta-ticket	10.0	0.3	SGD

A.3.2 Specific hyperparameters to Meta-ticket

- **Initial sparsity** For the initial sparsity $p_{\mathrm{init}} \in [0, 1]$, we chose $p_{\mathrm{init}} = 0.0$ (i.e. the initial subnetwork is equal to the entire network) since we found that a lower initial sparsity tends to be better in terms of meta-generalization ability. However, as we can see in Section B.2, we can get a more sparse subnetwork if we use a larger initial sparsity.
- **Parameter initialization** In Meta-ticket, there are two initialization for the score parameter $\mathbf{s}$ and the network parameter ϕ_0. (See Section 3.1 for the notation.) We employed Kaiminig uniform initialization [1] for $\mathbf{s}$, and Kaiming normal initialization [1] for ϕ_0.
- **Optimizer** As a (meta-)optimizer for the score parameters of Meta-ticket, following the strong lottery ticket literature [51, 9], we employed stochastic gradient descent (SGD) with a cosine scheduler. Also, we found that the outer learning rate needs to be larger than the standard ones, and thus we set the learning rate as 10.0. (See also Section B.1.)
- **Iterative randomization** Iterative randomization (IteRand, proposed by Chijiwa et al. [9]) is a technique to boost the performance of weight-pruning optimization, especially for small neural networks, by re-initializing the pruned parameters every K iterations. We chose the re-initialization frequency as $K = 1000$.

A.4 Implementations and training details

Implementations We implemented Meta-ticket and all experiments by using the PyTorch [48], learn2learn [4] and Torchmeta [10] libraries. Also, the implementation of ResNet-12 is based on the one implemented by Oh et al. [44].

Computational resources In meta-training and meta-testing, we used a single NVIDIA V100 GPU or NVIDIA A100 GPU for each experiment. For all of our experimental results, we reported means and one standard deviations for three random seeds.

Computational overhead of Meta-ticket Even though Meta-ticket has additional parameters for scores compared to MAML, there was little difference in meta-training time. For example, in the meta-training on miniImageNet with ResNet12 (Section 4.2), Meta-ticket takes about 583 seconds for 1000 iterations on an A100 GPU machine, while MAML takes about 560 seconds. Hence the computational overhead in this case is only about 4%.

B Additional experiments

B.1 Learning rates for Meta-ticket

We searched the outer learning rate for the score parameter of Meta-ticket, using the 1-shot 5-way benchmark on miniImageNet with ResNet-12. Table 7 shows the meta-validation accuracies for various learning rates. In contrast to standard training, relatively large learning rates are suitable for the score parameter $\mathbf{s} = (s_i)_{1 \leq i \leq N}$. This is because the actual value of each s_i is not important and just whether or not s_i is above the threshold σ matters.

Table 7: Meta-validation accuracies for various learning rates on the 1-shot 5-way miniImageNet benchmark with ResNet-12.

Learning rate	0.01	0.1	1.0	10.0	100.0
Accuracy	$34.87 \pm 1.40\%$	$42.97 \pm 0.67\%$	$53.67 \pm 2.12\%$	$\mathbf{54.30} \pm 2.52\%$	$51.97 \pm 0.64\%$

B.2 Effects of the initial sparsity

In this section, we analyze the effects of the initial sparsity p_{init} to the resulting subnetworks. Figure 5 shows the sparsity of the subnetwork in ResNet-12 obtained by Meta-ticket during the meta-training phase. Although the final sparsity largely depends on the initial sparsity, the sparsity changes in the direction of the half sparsity, consistently in every case. On the other hand, from the viewpoint of meta-generalization, we can see that the meta-validation accuracy tends to be better if we start from a lower initial sparsity (Table 8). Also, in Figure 6, we plotted the meta-validation accuracy curves during meta-training for each initial sparsity.

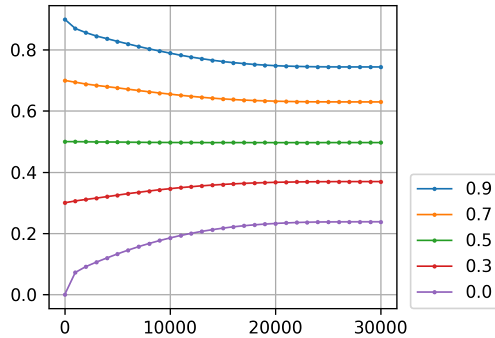

Figure 5: For each initial sparsity ($p_{\text{init}} = 0.0, 0.3, 0.5, 0.7, 0.9$), we plotted the sparsity of the subnetwork in ResNet-12 obtained by Meta-ticket during the meta-training phase. The x-axis is the number of meta-training iterations.

Table 8: Meta-validation accuracies for various initial sparsities in 1-shot 5-way setting.

Initial sparsity	0.0	0.3	0.5	0.7	0.9
Accuracy	$\mathbf{54.70} \pm 1.90\%$	$53.40 \pm 0.50\%$	$52.27 \pm 0.84\%$	$51.57 \pm 0.64\%$	$51.53 \pm 0.92\%$

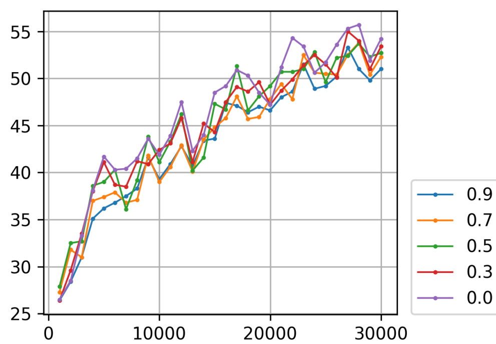

Figure 6: For each initial sparsity ($p_{\text{init}} = 0.0, 0.3, 0.5, 0.7, 0.9$), we plotted meta-validation accuracies during meta-training. The x-axis is the number of meta-training iterations.

B.3 Cross-domain evaluation in 1-shot 5-way setting

Table 9 shows the results of cross-domain evaluation on miniImageNet with the setting of 1-shot learning. We meta-trained ResNet-12 and WideResNet-28-10 by Meta-ticket and MAML, with the 1-shot learning tasks from the miniImageNet dataset. There seems to be little difference between the results of MAML and Meta-ticket (except for the case of WideResNet-28-10 evaluated on miniImageNet itself), in contrast to the 5-shot setting (Section 4.2). The results may be due to the lack of training samples during the inner optimization in the 1-shot setting, where we cannot take enough advantage of the rapid learning nature of Meta-ticket.

B.4 Comparison with state-of-the-art methods

In Table 10, we compare our cross-domain evaluation results (given in Table 2 in Section 4.2) to state-of-the-art methods (MetaOptNet [32] and Feature-wise Transformation [57]) other than MAML-based methods. Both of these state-of-the-art methods achieve higher meta-test accuracy on

Table 9: Cross-domain 1-shot 5-way evaluation with ResNet-12 and WideResNet-28-10

Networks	ResNet-12			WideResNet-28-10		
Meta-train	miniImageNet			miniImageNet		
Meta-test	miniImageNet	CUB	Cars	miniImageNet	CUB	Cars
MAML	$56.25 \pm 1.28\%$	$45.85 \pm 0.77\%$	$35.85 \pm 1.18\%$	$50.59 \pm 0.54\%$	$42.12 \pm 0.57\%$	$33.50 \pm 0.88\%$
Meta-ticket	$56.17 \pm 0.91\%$	$45.95 \pm 0.81\%$	$35.99 \pm 2.00\%$	$54.12 \pm 1.24\%$	$41.77 \pm 0.92\%$	$34.26 \pm 0.31\%$

miniImageNet, which is the dataset used in meta-training, than variants of MAML and Meta-ticket. This would be because these methods can leverage their strong feature extractor on the meta-training dataset. However, these two state-of-the-art methods are largely degraded when meta-tested on CUB and Stanford Cars. On the other hand, Meta-ticket + BOIL achieves similar accuracy as the feature-wise transformation method on CUB, and the highest accuracy on Stanford Cars in the table. This would show the strength of the rapid learning nature of Meta-ticket.

Table 10: Comparison with state-of-the-art methods in 5-shot 5-way cross-domain classification.

Meta-training dataset	miniImageNet		
Meta-test dataset	miniImageNet	CUB	Cars
MAML	$67.47 \pm 1.31\%$	$54.44 \pm 0.23\%$	$43.68 \pm 1.44\%$
ANIL	$66.88 \pm 1.59\%$	$53.90 \pm 1.17\%$	$40.87 \pm 3.95\%$
BOIL	$69.67 \pm 0.66\%$	$58.79 \pm 1.48\%$	$47.11 \pm 1.10\%$
Meta-ticket (Ours)	$71.31 \pm 0.29\%$	$57.97 \pm 0.53\%$	$45.90 \pm 0.50\%$
+ BOIL (Ours)	$74.23 \pm 0.30\%$	$64.06 \pm 1.05\%$	$\mathbf{55.20 \pm 0.64\%}$
MetaOptNet-SVM-trainval [32]	$80.00 \pm 0.45\%$[§]	$54.67 \pm 0.56\%$[§]	$45.90 \pm 0.49\%$[§]
GNN + Feature-wise Transformation [57]	$\mathbf{81.98 \pm 0.55\%}$[§]	$\mathbf{66.98 \pm 0.68\%}$[§]	$44.90 \pm 0.64\%$[§]

B.5 Results on specific to general/specific adaptation

In Section 4.2, we evaluated the cross-domain adaptation from a general-domain dataset (miniImageNet) to specific-domain datasets (CUB and Cars). Here we present additional experimental results (Table 11) of cross-domain adaptation from a specific-domain dataset (CUB) to the other datasets. Although there are only small difference between MAML-based methods and Meta-ticket when evaluated on the meta-training dataset itself, Meta-ticket has a larger gain on the specific to general/specific cross-domin adaptation. The results indicate that, while MAML-based methods successfully encode useful features into their initial parameters to classify the fine-grained classes of bird species (in CUB), the encoded features are not enough useful for classifying other categories in miniImageNet and Cars datasets.

Table 11: Results on specific to general/specific adaptation.

Meta-training dataset	CUB		
Meta-test dataset	CUB	miniImageNet	Cars
MAML	$78.92 \pm 0.62\%$	$43.03 \pm 0.26\%$	$38.95 \pm 0.42\%$
BOIL	$\mathbf{83.70 \pm 0.40\%}$	$49.17 \pm 1.30\%$	$43.93 \pm 1.39\%$
Meta-ticket	$80.49 \pm 0.50\%$	$46.01 \pm 0.55\%$	$40.24 \pm 0.92\%$
+ BOIL	$83.28 \pm 0.44\%$	$\mathbf{53.82 \pm 0.92\%}$	$\mathbf{48.85 \pm 0.56\%}$

B.6 Detailed plots of inner gradients during meta-training

In Section 3.2, we presented the plots of inner gradient norms of the last layer of the feature extractor of 5-MLP during meta-training on CIFAR-FS. Here we provide more detailed plots for every feature extracting layer of 5-MLP on CIFAR-FS (Figure 7) and VGG-Flower (Figure 8) with log-scaled

[§] These results are cited from Tseng et al. [57]

y-axis. In both cases, the inner gradient norms in MAML tend to converge to nearly zero, while the ones in Meta-ticket stop to decrease or even start to increase at some iteration. However, there are some exceptions particularly when inner learning rate is relatively large. This indicates that our theoretical discussion for a small inner learning rate (given in Section 3.2) does not necessarily describe the dynamics of inner gradients for large inner learning rates.

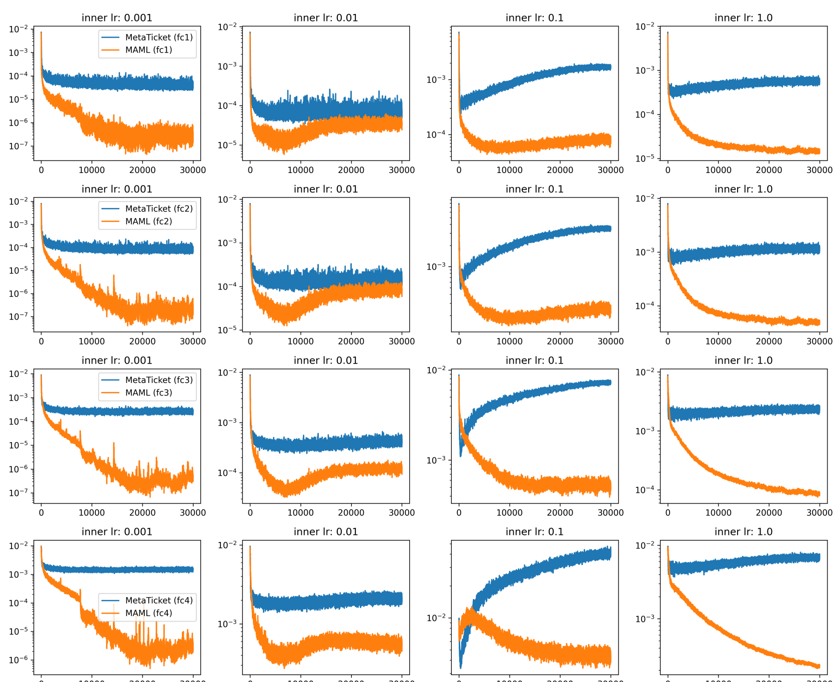

Figure 7: Inner gradient norms (log scale) of 5-MLP meta-trained on CIFAR-FS with various inner learning rates $\alpha \in \{0.001, 0.01, 0.1, 1.0\}$ for each fully-connected layer of the feature extractor.

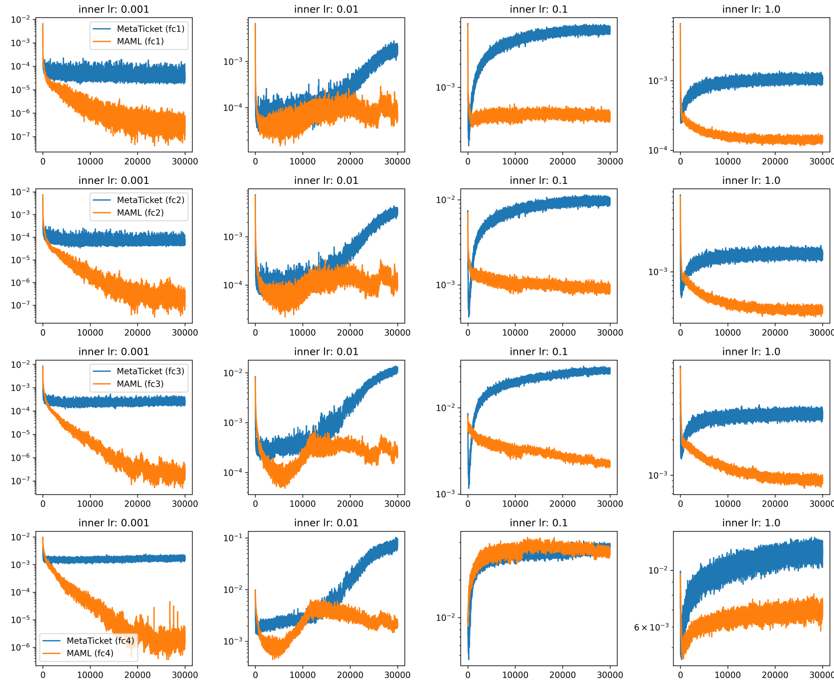

Figure 8: Inner gradient norms (log scale) of 5-MLP meta-trained on VGG-Flower with various inner learning rates $\alpha \in \{0.001, 0.01, 0.1, 1.0\}$ for each fully-connected layer of the feature extractor.

C Experiments on a regression benchmark

In this section, we report the results on a toy regression benchmark of learning to predict a given sine function, which is called Sinusoid regression [13], in the setting of 5-shot learning with 5 gradient steps. We used a simple 3-layered ReLU multilayered perceptron (MLP) with 1-dimensional input/output and 40-dimensional hidden layers, following the setting in Finn et al. [13]. First of all, we can predict that the naive application of Meta-ticket to the regression problem should fail because Meta-ticket cannot meta-learn the output scale of the neural network, in contrast to MAML which meta-learns the scale by meta-optimizing the NN parameter. Moreover, since the input/output of the network is 1-dimensional, pruning the input/output layer just decreases the hidden dimension after/before the input/output layer. Indeed, the mean squared error (MSE) loss of the naive application of Meta-ticket is only 3.56 ± 0.11, while MAML achieves 0.346 ± 0.113. Therefore, instead of the naive application, we apply Meta-ticket to the regression benchmark with the following configuration: For the input and output linear layers, instead of applying Meta-ticket, we simply meta-optimize the initial parameters for these layers in the same way as MAML. For the intermediate layer, we apply Meta-ticket and thus meta-optimize the sparse structure of the 40×40 matrix.

As a result, we observed that the modified application of Meta-ticket achieves the MSE loss of 0.596 ± 0.173, which is more comparable to MAML than the naive application. However, there still remains a gap between Meta-ticket and MAML in this benchmark. We consider that this may be because the direct parameter optimization (MAML) is more suitable for the simple functional approximation task than the meta-learned sparse structures (Meta-ticket).

References

[1] torch.nn.init – PyTorch 1.11.0 documentation. https://pytorch.org/docs/stable/nn.init.html. Accessed: 2022-05-01.

[2] Milad Alizadeh, Shyam A. Tailor, Luisa M Zintgraf, Joost van Amersfoort, Sebastian Farquhar, Nicholas Donald Lane, and Yarin Gal. Prospect pruning: Finding trainable weights at initialization using meta-gradients. In International Conference on Learning Representations, 2022.

[3] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems, 29, 2016.

[4] Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for Meta-Learning research. August 2020.

[5] Sungyong Baik, Seokil Hong, and Kyoung Mu Lee. Learning to forget for meta-learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2379–2387, 2020.

[6] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. Learning a synaptic learning rule. Université de Montréal, Département d'informatique et de recherche opérationnelle, 1990.

[7] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432, 2013.

[8] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In International Conference on Learning Representations, 2019.

[9] Daiki Chijiwa, Shin'ya Yamaguchi, Yasutoshi Ida, Kenji Umakoshi, and Tomohiro Inoue. Pruning randomly initialized neural networks with iterative randomization. Advances in Neural Information Processing Systems, 34, 2021.

[10] Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. Torchmeta: A Meta-Learning library for PyTorch, 2019. Available at: https://github.com/tristandeleu/pytorch-meta.

[11] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In International Conference on Learning Representations, 2020.

[12] Thomas Elsken, Benedikt Staffler, Jan Hendrik Metzen, and Frank Hutter. Meta-learning of neural architectures for few-shot learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12365–12375, 2020.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1126–1135. PMLR, 06–11 Aug 2017.

[14] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In International Conference on Learning Representations, 2018.

[15] Sebastian Flennerhag, Andrei A. Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. In International Conference on Learning Representations, 2020.

[16] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In International Conference on Learning Representations, 2019.

[17] Adam Gaier and David Ha. Weight agnostic neural networks. Advances in neural information processing systems, 32, 2019.

[18] Dawei Gao, Yuexiang Xie, Zimu Zhou, Zhen Wang, Yaliang Li, and Bolin Ding. Finding meta winning ticket to train your maml. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 411–420, 2022.

[19] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In International Conference on Machine Learning, pages 1704–1713. PMLR, 2018.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778. IEEE Computer Society, 2016.

[21] Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. arXiv preprint arXiv:1802.04376, 2018.

[22] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In International Conference on Artificial Neural Networks, pages 87–94. Springer, 2001.

[23] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. Advances in Neural Information Processing Systems, 32, 2019.

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pages 448–456. PMLR, 2015.

[25] Jaehong Kim, Sangyeul Lee, Sungwan Kim, Moonsu Cha, Jung Kwon Lee, Youngduck Choi, Yongseok Choi, Dong-Yeon Cho, and Jiwon Kim. Auto-meta: Automated gradient based meta learner search. arXiv preprint arXiv:1806.06927, 2018.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations, 2015.

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In Proceedings of the IEEE international conference on computer vision workshops, pages 554–561, 2013.

[28] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[29] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015.

[30] Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In International Conference on Learning Representations, 2020.

[31] Hayeon Lee, Eunyoung Hyung, and Sung Ju Hwang. Rapid neural architecture search by learning to generate graphs from datasets. In International Conference on Learning Representations, 2020.

[32] 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.

[33] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In International Conference on Learning Representations, 2019.

[34] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. arXiv preprint arXiv:1707.09835, 2017.

[35] Dongze Lian, Yin Zheng, Yintao Xu, Yanxiong Lu, Leyu Lin, Peilin Zhao, Junzhou Huang, and Shenghua Gao. Towards fast adaptation of neural architectures with meta learning. In International Conference on Learning Representations, 2020.

[36] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In International Conference on Learning Representations, 2019.

[37] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013.

[38] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In International Conference on Machine Learning, pages 6682–6691. PMLR, 2020.

[39] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Proceedings of the European Conference on Computer Vision (ECCV), pages 67–82, 2018.

[40] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 7765–7773, 2018.

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

[42] Behnam Neyshabur. Towards learning convolutions from scratch. Advances in Neural Information Processing Systems, 33:8078–8088, 2020.

[43] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729. IEEE, 2008.

[44] Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. Boil: Towards representation change for few-shot learning. In International Conference on Learning Representations, 2021.

[45] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 2925–2934. Curran Associates, Inc., 2020.

[46] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. Neural Networks, 113:54–71, 2019.

[47] Eunbyung Park and Junier B Oliva. Meta-curvature. Advances in Neural Information Processing Systems, 32, 2019.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

[49] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 2599–2610. Curran Associates, Inc., 2020.

[50] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In International Conference on Learning Representations, 2020.

[51] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11893–11902, 2020.

[52] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In International Conference on Learning Representations, 2017.

[53] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[54] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. Neural Computation, 4(1):131–139, 1992.

[55] Sebastian Thrun. Lifelong learning algorithms. In Learning to learn, pages 181–209. Springer, 1998.

[56] Hongduan Tian, Bo Liu, Xiao-Tong Yuan, and Qingshan Liu. Meta-learning with network pruning. In European Conference on Computer Vision, pages 675–700. Springer, 2020.

[57] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In International Conference on Learning Representations, 2020.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

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

[60] Johannes Von Oswald, Dominic Zhao, Seijin Kobayashi, Simon Schug, Massimo Caccia, Nicolas Zucchet, and João Sacramento. Learning where to learn: Gradient sparsity in meta and continual learning. Advances in Neural Information Processing Systems, 34, 2021.

[61] Haoxiang Wang, Yite Wang, Ruoyu Sun, and Bo Li. Global convergence of maml and theory-inspired neural architecture search for few-shot learning. arXiv preprint arXiv:2203.09137, 2022.

[62] Jiaxing Wang, Jiaxiang Wu, Haoli Bai, and Jian Cheng. M-nas: Meta neural architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6186–6193, 2020.

[63] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. California Institute of Technology, CNS-TR-2010-001, 2010.

[64] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. Advances in Neural Information Processing Systems, 33:15173–15184, 2020.

[65] Haoran You, Baopu Li, Zhanyi Sun, Xu Ouyang, and Yingyan Lin. Supertickets: Drawing task-agnostic lottery tickets from supernets via jointly architecture searching and parameter pruning. In European Conference on Computer Vision, pages 674–690. Springer, 2022.

[66] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 87.1–87.12. BMVA Press, September 2016.

[67] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In International Conference on Learning Representations, 2017.

[68] Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization. In International Conference on Learning Representations, 2021.

[69] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[70] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In International Conference on Learning Representations, 2017.