Data Augmentation for Meta-Learning

Renkun Ni
University of Maryland
rn9zm@cs.umd.edu

Micah Goldblum
University of Maryland
goldblum@umd.edu

Amr Sharaf
University of Maryland
amr@cs.umd.edu

Kezhi Kong
University of Maryland
kong@cs.umd.edu

Tom Goldstein
University of Maryland
tomg@cs.umd.edu

Abstract

Conventional image classifiers are trained by randomly sampling mini-batches of images. To achieve state-of-the-art performance, sophisticated data augmentation schemes are used to expand the amount of training data available for sampling. In contrast, meta-learning algorithms sample not only images, but classes as well. We investigate how data augmentation can be used not only to expand the number of images available per class, but also to generate entirely new classes. We systematically dissect the meta-learning pipeline and investigate the distinct ways in which data augmentation can be integrated at both the image and class levels. Our proposed meta-specific data augmentation significantly improves the performance of meta-learners on few-shot classification benchmarks.

1 Introduction

Data augmentation has become an essential part of the training pipeline for image classifiers and related tasks, as it offers a simple and efficient way to significantly improve performance [6, 26]. In contrast, little work exists on data augmentation for meta-learning. Existing frameworks for few-shot image classification use only horizontal flips, random crops, and color jitter to augment images in a way that parallels augmentation for conventional training [2, 14]. Meanwhile, meta-learning methods have received increasing attention as they have reached the cutting edge of few-shot performance. While new meta-learning algorithms emerge at a rapid rate, we show that, like image classifiers, meta-learners can achieve significant performance boosts through carefully chosen data augmentation strategies that are injected into the various stages of the meta-learning pipeline.

Meta-learning frameworks use data for multiple purposes during each gradient update, which creates the possibility for a diverse range of data augmentations that are not possible within the standard training pipeline. We explore these possibilities and discover combinations of augmentation types that improve performance over existing methods. Our contributions can be summarized as follows:

- First, we break down the meta-learning pipeline and identify places in which data augmentation can be inserted. We uncover four modes of augmentations for meta-learning: support augmentation, query augmentation, task augmentation, and shot augmentation.
- Second, we test these four modes using a pool of image augmentations, and we find that query augmentation is critical, while support augmentations often do not provide performance benefits and may even degrade accuracy in some cases.
- Third, we combine augmentations and implement a MaxUp strategy, which we call Meta-MaxUp, in order to maximize performance. We achieve significant performance boosts with popular meta-learners on both mini-ImageNet and CIFAR-FS.

4th Workshop on Meta-Learning at NeurIPS 2020, Vancouver, Canada.

2 Related Work

Meta-learners are known to be particularly vulnerable to overfitting [18]. One recent work has developed a data augmentation method to overcome this problem [15]. The latter method involves simply rotating all images in a class by a large amount and considering this new rotated class to be distinct from its parent class. This effectively increases the number of possible few-shot tasks that can be sampled during training. A feature-space augmentation, MetaMix, has been proposed for averaging support features in few-shot learning [23]. A different line of work has instead applied regularizers to prevent overfitting and improve few-shot classification [9, 24]. Yet additional work has developed methods for labeling and augmenting unlabeled data [1, 4], generative models for deforming images in one-shot metric learning [5], and feature space data augmentation for adapting language models to new unseen intents [12].

3 The Anatomy of Data Augmentation for Meta-Learning

Adopting common terminology from the literature, the archetypal meta-learning algorithm contains an *inner loop* and an *outer loop* in each parameter update of the training procedure. During an episode of training, we sample a batch of tasks which may be, for example, five-way classification problems. In the inner loop, a model is fine-tuned or adapted on *support* data. Then, in the outer loop, the model is evaluated on *query* data, and we compute the gradient of the loss on the query data with respect to the model's parameters before fine-tuning. Finally, we perform a descent step, completing the episode. Intuitively, meta-learners are optimized to generalize well after fine-tuning on very little data. At test time, the model is fine-tuned on a small set of data, which is analogous to support data, and then inference is performed on other data, analogous to query data. The number of support samples per class in a few-shot classification problem is called the *shot*.

3.1 Data Augmentation Modes

We describe four modes of data augmentation for meta-learning which may be employed individually or combined.

Support augmentation: Data augmentation may be applied to support data in the inner loop of fine-tuning. This strategy enlarges the pool of fine-tuning data.

Query augmentation: Data augmentation alternatively may be applied to query data. This strategy enlarges the pool of evaluation data to be sampled during training.

Task augmentation: We can increase the number of possible tasks by uniformly augmenting whole classes to add new classes with which to train. For example, a vertical flip applied to all car images yields a new upside-down car class which may be sampled during training.

Shot augmentation: At test time, we can artificially amplify the shot by adding additional copies of each image using data augmentation. Shot augmentation can also be used during training by adding copies of each support image via augmentation. Shot augmentation during training may better prepare meta-learners for test-time shot augmentation.

Existing meta-learning algorithms for few-shot image classification typically use horizontal flips, random crops, and color jitter on both support and query images. In Section 4, we test the four modes of data augmentation enumerated above in isolation across a large array of specific augmentations. We find that query augmentation is far more critical than support augmentation for increasing performance. Additionally, we find that task augmentation, when combined with query augmentation, can offer further boosts in performance when compared with existing frameworks.

3.2 Data Augmentation Techniques

For each of the data augmentation modes described above, we try a variety of specific data augmentation techniques. Some techniques are only applicable to support, query, and shot modes or solely to the task mode. We use an array of standard augmentation techniques as well as CutMix [25],

MixUp [26], and Self-Mix [20]. In the context of the task augmentation mode, we apply these the same way to every image in a class in order to augment the number of classes. For example, we use MixUp to create a half dog half truck class where every image is the average of a dog image and a truck image. We also try combining multiple classes into one class as a task augmentation mode. In general, techniques which greatly change the image distribution are better suited for task augmentations while techniques that preserve the image distribution are typically better suited for the support, query, and shot augmentation modes. The baseline models we compare to use horizontal flip, random crop, and color jitter augmentation techniques at both the support and query levels since these techniques are prevalent in the literature. More details on our pool of augmentation techniques can be found in Appendix A.1.

3.3 Meta-MaxUp Augmentation for Meta-Learning

Recent work proposed MaxUp augmentation to alleviate overfitting during the training of classifiers [10]. This strategy applies many augmentations to each image and chooses the augmented image which yields the highest loss. MaxUp is conceptually similar to adversarial training [16]. Like adversarial training, MaxUp involves solving a saddlepoint problem in which loss is minimized with respect to parameters while being maximized with respect to the input. In the standard image classification setting, MaxUp, together with CutMix, improves generalization and achieves state-of-the-art performance on ImageNet. Here, we extend MaxUp to the setting of meta-learning. Before training, a set of the data augmentations, $\mathcal{S}$, collected from the four modes, as well as their combinations, is chosen. For example, $\mathcal{S}$ may contain horizontal flip shot augmentation, query CutMix, and the combination of both. During each iteration of training, we first sample a batch of tasks, each containing support and query data, as is typical in the meta-learning framework. For each element in the batch, we randomly select m augmentations from the set $\mathcal{S}$, and we apply these to the task, generating m augmented tasks with augmented support and query data. Then, for each element of the batch of tasks originally sampled, we choose the augmented task that maximizes loss, and we perform a parameter update step to minimize training loss. Formally, we solve the minimax optimization problem,

$$\min_{\theta} \mathbb{E}_{\mathcal{T}} \Big[\max_{M \in \mathcal{S}} \mathcal{L}(F_{\theta'}, M(\mathcal{T}^q)) \Big], \tag{1}$$

where $\theta' = \mathcal{A}(\theta, M(\mathcal{T}^s))$, $\mathcal{A}$ denotes fine-tuning, F is the base model with parameters θ, $\mathcal{L}$ is the loss function used in the outer loop of training, and $\mathcal{T}$ is a task with support and query data $\mathcal{T}^s$ and $\mathcal{T}^q$, respectively. A detailed algorithm can be found in A.2.

4 Experiments

In this section, we empirically demonstrate the following:

1. Augmentations applied in the four distinct modes behave differently. In particular, query and task augmentation are far more important than support augmentation. (Section 4.2)
2. Meta-specific data augmentation strategies can improve performance over the generic strategies commonly used for meta-learning. (Section 4.2)
3. We can further boost performance by combining augmentations with Meta-MaxUp. (Section 4.3)

4.1 Experimental Setup

We conduct experiments on four meta-learning algorithms: ProtoNet [21], R2-D2 [2], MetaOptNet [14], and MCT [13]. ProtoNet is a metric-learning method that uses a prototype learning head, which classifies samples by extracting a feature vector and then performing a nearest-neighbor search for the closest class prototype. R2-D2 and MetaOptNet instead use differentiable solvers with a ridge regression and SVM head, respectively. These methods extract feature vectors and then apply a standard linear classifer to assign class labels. MCT improves upon ProtoNet by meta-learning confidence scores. In the main body, we report the results of all these different classifier head options, all using the ResNet-12 backbone proposed by [17]. The results of the original backbone can be found in Appendix A.6.

Table 1: Few-shot classification accuracy (%) on the CIFAR-FS dataset with the most effective data augmentations for each mode shown. "CNN-4" denotes a 4-layer convolutional network with 96, 192, 384, and 512 filters in each layer [2]. Best performance in each category is bolded.

		CNN-4		ResNet-12	
Mode	Level	1-shot	5-shot	1-shot	5-shot
Baseline	-	67.56 ± 0.35	82.39 ± 0.26	73.01 ± 0.37	84.29 ± 0.24
Self-Mix	Support	69.61 ± 0.35	83.43 ± 0.25	71.96 ± 0.36	84.84 ± 0.25
CutMix	Query	$\mathbf{70.54 \pm 0.33}$	$\mathbf{84.69 \pm 0.24}$	$\mathbf{75.97 \pm 0.34}$	$\mathbf{87.28 \pm 0.23}$
Large Rotation	Task	68.96 ± 0.35	83.65 ± 0.25	73.79 ± 0.36	85.81 ± 0.24
Horizontal Flip	Shot	68.13 ± 0.35	82.95 ± 0.25	73.25 ± 0.36	85.06 ± 0.25

We perform our experiments on the mini-ImageNet and CIFAR-FS datasets [2, 22]. Mini-ImageNet is a few-shot learning dataset derived from the ImageNet classification dataset [7], and CIFAR-FS is derived from CIFAR-100 [11]. Across all our experiments, we consider the inductive setting for few-shot learning, in which each test image is evaluated separately and independently. For fair comparison, we only compare inductive methods to other inductive methods. A description of datasets and training hyperparameters can be found in Appendix A.4. We report confidence intervals with a radius of one standard error.

4.2 An Empirical Comparison of Augmentation Modes

We empirically evaluate the performance of all four different augmentation modes identified in Section 3.1 on the CIFAR-FS dataset using an R2-D2 base-learner paired with both a 4-layer CNN backbone (as used in the original work) and a ResNet-12 backbone. We report the results of the most effective augmentations for each modes in Table 1. The full table can be found in Appendix A.3.

Table 1 demonstrates that each mode of augmentation individually can improve performance. Augmentation applied to query data is consistently more effective than the other augmentation modes. In particular, simply applying CutMix to query samples improves accuracy by as much as 3% on both backbones. In contrast, most augmentations on support data actually damage performance. In addition, results renedred by combining query CutMix with other effective augmentations are displayed in Appendix A.5. Interestingly, when we use CutMix on both support and query images, we observe worse performance than simply using CutMix on query data alone. Therefore, existing meta-learning methods, which apply the same augmentations to query and support data without using task and shot augmentation, may be achieving suboptimal performance. In order to further boost performance, we propose Meta-MaxUp for combining various augmentations in different modes.

4.3 Meta-MaxUp Further Improves Performance

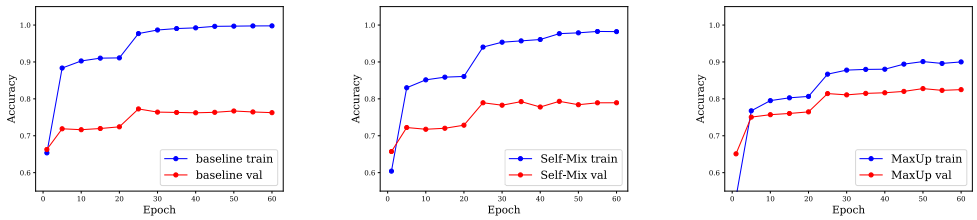
In this section, we evaluate our proposed Meta-MaxUp strategy in the same experimental setting as above for various values of m and different data augmentation pool sizes. Results with best m are reported in Table 2, and a detailed description of the augmentation pools as well as the full results can be found in Appendix A.7. Rows beginning with "CutMix" denote experiments in which the pool of augmentations simply includes many CutMix samples. "Single" denotes experiments in which each augmentation in $\mathcal{S}$ is of a single type, while "Medium" and "Large" denote experiments in which each element of $\mathcal{S}$ is a combination of augmentations, for example CutMix+rotation. Combinations greatly expand the number of augmentations in the pool. As we increase m and include a large number of augmentations in the pool, we observe performance boosts as high as 4% over the baseline, which uses horizontal flip, random crop, and color jitter data augmentations from the original work corresponding to the R2-D2 meta-learner used [2].

We explore the training benefits of these meta-specific training schemes by examining saturation during training. To this end, we plot the training and validation accuracy over time for R2-D2 meta-learners with ResNet-12 backbones using baseline augmentations, query Self-Mix, and Meta-MaxUp with a medium sized pool and $m = 4$. See Figure 1 for training and validation accuracy curves. With only baseline augmentations, validation accuracy stops increasing immediately after the first learning rate decay. This suggests that baseline augmentations do not prevent overfitting during meta-training. In contrast, we observe that models trained with Meta-MaxUp do not quickly overfit and continue

Table 2: Few-shot classification accuracy (%) on the CIFAR-FS dataset for Meta-MaxUp over different sizes of augmentation pools with number of samples $m = 4$.

		CNN-4		ResNet-12	
Pool	m	1-shot	5-shot	1-shot	5-shot
Baseline	-	67.56 ± 0.36	82.39 ± 0.26	73.01 ± 0.37	84.29 ± 0.24
CutMix	4	70.48 ± 0.34	84.76 ± 0.24	75.08 ± 0.23	87.60 ± 0.24
Single	4	$\mathbf{71.10 \pm 0.34}$	$\mathbf{85.50 \pm 0.24}$	76.82 ± 0.24	88.14 ± 0.23
Medium	4	70.58 ± 0.34	85.32 ± 0.24	76.30 ± 0.24	88.29 ± 0.22
Large	4	70.71 ± 0.34	85.04 ± 0.23	$\mathbf{76.99 \pm 0.24}$	$\mathbf{88.35 \pm 0.22}$

improving validation performance for a greater number of epochs. Meta-MaxUp visibly reduces the generalization gap.



(a) (b) (c)

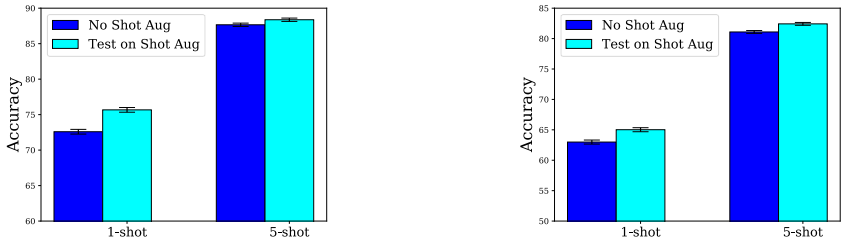
Figure 1: Training and validation accuracy for R2-D2 meta-learner with ResNet-12 backbone on the CIFAR-FS dataset. (a) Baseline model (b) query Self-Mix (c) Meta-MaxUp.

4.4 Shot Augmentation for Pre-Trained Models

In the typical meta-learning framework, data augmentations are used during meta-training but not during test time. On the other hand, in some transfer learning work, data augmentations, such as horizontal flips, random crops, and color jitter, are used during fine-tuning at test time [3]. These techniques enable the network to see more data samples during few-shot testing, leading to enhanced performance. We propose shot augmentation (see Section 3) to enlarge the number of few-shot samples during testing, and we also propose a variant in which we additionally train using the same augmentations on support data in order to prepare the meta-learner for this test time scenario. Figure 2 shows the effect of shot augmentation (using only horizontal flips) on performance for MetaOptNet with ResNet-12 backbone trained with Meta-MaxUp. Shot augmentation consistently improves results across datasets, especially on 1-shot classification ($\sim 2\%$). To be clear, in this figure, we are not using shot augmentation during the training stage. Rather, we are using conventional low-shot training, and then deploying our models with shot augmentation at test time. These post-training performance gains can be achieved by directly applying shot augmentation on pre-trained models during testing. For additional experiments, see Appendix A.8.

4.5 Improving Existing Meta-Learners with Better Data Augmentation

In this section, we improve the performance of four different popular meta-learning methods including ProtoNet [21], R2-D2 [2], MetaOptNet [14], and MCT [13]. We compare their baseline performance to query CutMix with task-level rotation as well as Meta-MaxUp data augmentation strategies on both the CIFAR-FS and mini-ImageNet datasets. See Table 3 for the results of these experiments. In all cases, we are able to improve the performance of existing methods, sometimes by over 5%. Even without Meta-MaxUp, we improve performance over the baseline by a large margin. The superiority of meta-learners that use these augmentation strategies suggests that data augmentation is critical for these popular algorithms and has largely been overlooked.



(a) (b)

Figure 2: Performance on shot augmentation using MetaOptNet trained with the proposed Meta-MaxUp. (a) 1-shot and 5-shot on CIFAR-FS (b) 1-shot and 5-shot on mini-ImageNet.

In addition, we compare our results with augmentation by Large Rotations at the task level – the only competing work to our knowledge – in Appendix A.9. Note, augmentation with Large Rotations to create new classes is referred to as "Task Augmentation" in [15]; we refer to it here as "Large Rotations" to avoid confusion since we study a myriad of augmentations at the task level. We observe that with the same training algorithm (MetaOptNet with SVM) and the ResNet-12 backbone, our method outperforms the Large Rotations augmentation strategy by a large margin on both the CIFAR-FS and mini-ImageNet datasets. Together with the same ensemble method as used in Large Rotations, marked by "+ens", we further boost performance consistently above the MCT baseline, the current highest performing meta-learning method on these benchmarks, despite using an older meta-learner previously thought to perform worse than MCT. Moreover, when both training and validation datasets are used for meta-training, we can achieve the state-of-art results for few-shot classification on mini-ImageNet dataset in inductive setting.

Table 3: Few-shot classification accuracy (%) on CIFAR-FS and mini-ImageNet with ResNet-12. "+ DA" denotes training with CutMix (Q) + Rotation (T), and "+ MM" denotes training with Meta-MaxUp.

	CIFAR-FS		mini-ImageNet	
Method	1-shot	5-shot	1-shot	5-shot
R2-D2	73.01 ± 0.37	84.29 ± 0.24	60.46 ± 0.32	76.88 ± 0.24
+ DA	76.17 ± 0.34	87.74 ± 0.24	**65.54 ± 0.32**	81.52 ± 0.23
+ MM	**76.65 ± 0.33**	**88.57 ± 0.24**	65.15 ± 0.32	**81.76 ± 0.24**
ProtoNet	70.21 ± 0.36	84.26 ± 0.25	57.34 ± 0.34	75.81 ± 0.25
+ DA	74.30 ± 0.36	86.24 ± 0.24	60.82 ± 0.34	78.23 ± 0.25
+ MM	**76.05 ± 0.34**	**87.84 ± 0.23**	**62.81 ± 0.34**	**79.38 ± 0.24**
MetaOptNet	70.99 ± 0.37	84.00 ± 0.25	60.01 ± 0.32	77.42 ± 0.23
+ DA	74.56 ± 0.34	87.61 ± 0.23	64.94 ± 0.33	82.10 ± 0.23
+ MM	**75.67 ± 0.34**	**88.37 ± 0.23**	**65.02 ± 0.32**	**82.42 ± 0.23**
MCT	75.80 ± 0.33	89.10 ± 0.42	64.84 ± 0.33	81.45 ± 0.23
+ MM	**76.00 ± 0.33**	**89.54 ± 0.33**	**66.37 ± 0.32**	**83.11 ± 0.22**

5 Discussion

In this work, we break down data augmentation in the context of meta-learning. In doing so, we uncover possibilities that do not exist in the classical image classification setting. We identify four modes of augmentation: query, support, task, and shot. These modes behave differently and are of varying importance. Specifically, we find that it is particularly important to augment query data. After adapting various data augmentations to meta-learning, we propose Meta-MaxUp for combining various meta-specific data augmentations. We demonstrate that Meta-MaxUp significantly improves the performance of popular meta-learning algorithms. We hope that this work opens up possibilities

for further work on meta-specific data augmentation and that emerging methods for data augmentation will boost the performance of meta-learning on large backbones without overfitting.

Acknowledgement

The university of Maryland team was supported by the ONR MURI program, AFOSR MURI program, and the National Science Foundation DMS division. Addition support was provided by DARPA GARD, DARPA QED4RML, and DARPA YFA.

References

[1] Antreas Antoniou and Amos Storkey. Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. *arXiv preprint arXiv:1902.09884*, 2019.

[2] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 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] Zitian Chen, Yanwei Fu, Kaiyu Chen, and Yu-Gang Jiang. Image block augmentation for one-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3379–3386, 2019.

[5] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8680–8689, 2019.

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Micah Goldblum, Liam Fowl, and Tom Goldstein. Adversarially robust few-shot learning: A meta-learning approach. *arXiv*, pages arXiv–1910, 2019.

[9] Micah Goldblum, Steven Reich, Liam Fowl, Renkun Ni, Valeriia Cherepanova, and Tom Goldstein. Unraveling meta-learning: Understanding feature representations for few-shot tasks. *arXiv preprint arXiv:2002.06753*, 2020.

[10] Chengyue Gong, Tongzheng Ren, Mao Ye, and Qiang Liu. Maxup: A simple way to improve generalization of neural network training. *arXiv preprint arXiv:2002.09024*, 2020.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. A closer look at feature space data augmentation for few-shot intent classification. *arXiv preprint arXiv:1910.04176*, 2019.

[13] Seong Min Kye, Hae Beom Lee, Hoirin Kim, and Sung Ju Hwang. Transductive few-shot learning with meta-learned confidence. *arXiv preprint arXiv:2002.12017*, 2020.

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

[15] Jialin Liu, Fei Chao, and Chih-Min Lin. Task augmentation by rotating for meta-learning. *arXiv preprint arXiv:2003.00804*, 2020.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[17] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018.

[18] Janarthanan Rajendran, Alex Irpan, and Eric Jang. Meta-learning requires meta-augmentation. *arXiv preprint arXiv:2007.05549*, 2020.

[19] 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.

[20] Jin-Woo Seo, Hong-Gyu Jung, and Seong-Whan Lee. Self-augmentation: Generalizing deep networks to unseen classes for few-shot learning. *arXiv preprint arXiv:2004.00251*, 2020.

[21] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

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

[23] Huaxiu Yao, Longkai Huang, Ying Wei, Li Tian, Junzhou Huang, and Zhenhui Li. Don't overlook the support set: Towards improving generalization in meta-learning. *arXiv preprint arXiv:2007.13040*, 2020.

[24] Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. *arXiv preprint arXiv:1912.03820*, 2019.

[25] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.