A. Impact of Dataset Diversity on Various Stages of Meta-learning

Following the settings in Section 3.1, we further investigate how dataset diversity matters in the various stages of meta-learning on Mini-ImageNet, CNN-4 backbones, and with the ProtoNet (Snell et al., 2017) head. The results are shown in Table 8, and all these results support our findings that meta learning algorithms are more sensitive to the amount of query data and number of tasks and less sensitive to the amount of support data.

Table 8. Few-shot classification accuracy (%) using different meta-learning algorithms and backbones for various data size manipulations on Mini-ImageNet. "Support", "Query" and "Task" columns denote the number of samples per class for support and query data and the number of total tasks available for sampling. Confidence intervals have radius equal to one standard error.

Method	Backbone	Support	Query	Task	1-shot	5-shot
R2-D2	CNN-4	600	600	full	55.94 ± 0.32	72.32 ± 0.25
R2-D2	CNN-4	5	600	full	55.05 ± 0.31	71.66 ± 0.26
R2-D2	CNN-4	5 (random)	600	full	42.09 ± 0.29	60.12 ± 0.27
R2-D2	CNN-4	600	5	full	49.87 ± 0.30	66.00 ± 0.27
R2-D2	CNN-4	600	600	13	53.34 ± 0.31	69.43 ± 0.25
R2-D2	ResNet-12	600	600	full	60.50 ± 0.33	76.60 ± 0.24
R2-D2	ResNet-12	5	600	full	58.79 ± 0.32	76.45 ± 0.25
R2-D2	ResNet-12	5 (random)	600	full	43.80 ± 0.30	62.26 ± 0.28
R2-D2	ResNet-12	600	5	full	48.02 ± 0.31	65.45 ± 0.26
R2-D2	ResNet-12	600	600	13	57.65 ± 0.33	73.18 ± 0.28
ProtoNet	ResNet-12	600	600	full	57.46 ± 0.38	75.61 ± 0.29
ProtoNet	ResNet-12	5	600	full	57.03 ± 0.33	75.46 ± 0.26
ProtoNet	ResNet-12	5 (random)	600	full	43.36 ± 0.32	57.20 ± 0.28
ProtoNet	ResNet-12	600	5	full	48.40 ± 0.34	64.79 ± 0.29
ProtoNet	ResNet-12	600	600	13	51.88 ± 0.35	66.41 ± 0.29

B. Details About Data Augmentation Techniques

In this section, we provide more details about the different data augmentation techniques we use in this work. We employ the following pool of data augmentations:

CutMix: Yun et al. (2019) introduce the CutMix augmentation strategy where patches are cut and pasted among training images, and the ground truth labels are also mixed proportionally to the area of the patches.

MixUp: Zhang et al. (2017) propose mixup, a simple learning principle to alleviate memorization and sensitivity to adversarial examples. Mixup trains a neural network on convex combinations of pairs of examples and their labels. By doing so, mixup regularizes the neural network to favor simple linear behavior in between training examples.

Self-Mix: Seo et al. (2020) introduce the self-mix augmentation strategy in which a patch of an image is substituted into other values in the same image to improve the generalization ability of few-shot image classification models.

In addition, we use some standard and simple data augmentation techniques:

Rotation: augments the data by rotating the images.

Horizontal Flip: augments the data by horizontally flipping images.

Random Erase: augments the data by randomly erasing patches from the image.

Finally, we also experimented with the following data augmentation techniques:

Combining Labels: augments the data by combining two different labels into a single class. For instance, we may combine the “dog” and “cat” labels to create a new “dog or cat” class.

Feature Mixup: similar to the “Mixup” augmentation technique we describe above, however we perform the mixup strategy on the feature representation for the image.

Drop Channel: augments the data by dropping color channels in the image.

Solarize: inverts all pixels above a threshold value of magnitude.

C. Training Details

For MetaOptNet, we use the same training procedure as (Lee et al., 2019) including SGD with Nesterov momentum of 0.9 and weight decay coefficient 0.0005. The model was meta-trained for 60 epochs, with an initial learning rate 0.1, then changed to 0.006, 0.0012, and 0.00024 at epochs 20, 40 and 50, respectively. In each epoch, we train on 8000 episodes and use mini-batches of size 8. Following (Lee et al., 2019), we use a larger shot number (15) to train mini-ImageNet for both 1-shot and 5-shot classification. For MCT, we use the same optimizer but with batch size 1 and maximum iterations 50000. Following (Kye et al., 2020), we enlarge the training classification ways to 15 for a 5-way testing. We use instance-wise metric for all inductive learning.

Table 9 compares the training time of meta-learning methods with baseline data augmentations, with our proposed data augmentations (DA) and Meta-MaxUp strategy (MM) on the CIFAR-FS dataset. We employ data parallelism across 4 Nvidia RTX 2080 Ti GPUs for all experiments. The training time of meta algorithms with our proposed data augmentation is almost the same as with baseline methods. Although Meta-MaxUp requires m times as many forward passes, here $m = 4$, it does not require any extra backward passes. Thus, Meta-MaxUp typically runs roughly 2-3 times longer than baseline methods.

Table 9. Runtime (training time in hours for 60 epochs) comparison of data augmentation strategies on CIFAR-FS

Method	Backbone	Runtime	Backbone	Runtime
R2D2	CNN-4	2.5h	ResNet-12	3.2h
+ DA	CNN-4	2.6h	ResNet-12	3.7h
+ MM	CNN-4	4.1h	ResNet-12	8.2h
MetaOptNet	CNN-4	8.6h	ResNet-12	8.9h
+ DA	CNN-4	8.8h	ResNet-12	9.2h
+ MM	CNN-4	14.5h	ResNet-12	18.5h

D. Results for All Data Augmentation Techniques

Table 10 shows the few-shot classification accuracy on CIFAR-FS of an R2-D2 meta-learner with all single data augmentation techniques used in our paper. We highlight the best result in each mode. Data augmentation on query images significantly improves the baseline performance as well as data augmentations on other modes.

E. Results for Combination of Data Augmentations

Table 11 shows the few-shot classification accuracy for combinations of data augmentations building on the top of query CutMix, with both CNN-4 and ResNet-12 backbones.

Table 10. Few-shot classification accuracy (%) on the CIFAR-FS dataset for all data augmentations with an R2-D2 learner. Confidence intervals have radius equal to one standard error. "CNN-4" denotes a 4-layer convolutional network with 96, 192, 384, and 512 filters in each layer (Bertinetto et al., 2018). 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	71.95 ± 0.37	84.56 ± 0.25
Random Erase	Support	67.71 ± 0.36	82.25 ± 0.26	72.30 ± 0.37	84.50 ± 0.25
Self-Mix	Support	**69.61 ± 0.35**	**83.43 ± 0.25**	71.96 ± 0.36	**84.84 ± 0.25**
CutMix	Support	69.05 ± 0.36	83.12 ± 0.26	**72.79 ± 0.37**	84.70 ± 0.25
MixUp	Support	68.64 ± 0.37	82.72 ± 0.27	71.86 ± 0.37	84.11 ± 0.25
Feature Mixup	Support	67.88 ± 0.35	82.40 ± 0.25	71.21 ± 0.37	83.38 ± 0.25
Rotation	Support	68.65 ± 0.35	82.86 ± 0.25	71.13 ± 0.37	83.84 ± 0.25
Combining labels	Support	68.27 ± 0.36	82.53 ± 0.26	71.00 ± 0.38	83.12 ± 0.25
Drop Channel	Support	68.21 ± 0.35	82.76 ± 0.25	69.65 ± 0.73	83.15 ± 0.25
Solarize	Support	68.65 ± 0.35	82.68 ± 0.26	70.88 ± 0.37	83.45 ± 0.25
Random Erase	Query	69.73 ± 0.34	84.04 ± 0.25	73.05 ± 0.36	85.67 ± 0.25
Self-Mix	Query	69.55 ± 0.35	84.20 ± 0.25	73.59 ± 0.35	86.14 ± 0.25
CutMix	Query	**70.54 ± 0.33**	**84.69 ± 0.24**	**75.97 ± 0.34**	**87.28 ± 0.23**
MixUp	Query	67.70 ± 0.34	83.13 ± 0.25	72.93 ± 0.35	86.13 ± 0.24
Feature Mixup	Query	70.16 ± 0.35	83.80 ± 0.28	73.38 ± 0.35	85.87 ± 0.23
Rotation	Query	68.17 ± 0.35	83.01 ± 0.25	72.02 ± 0.36	84.42 ± 0.25
Combining labels	Query	66.01 ± 0.34	81.99 ± 0.26	69.77 ± 0.37	82.99 ± 0.26
Drop Channel	Query	68.34 ± 0.35	83.25 ± 0.25	69.60 ± 0.37	83.01 ± 0.26
Solarize	Query	67.51 ± 0.35	82.65 ± 0.25	72.45 ± 0.36	84.97 ± 0.24
MixUp	Task	67.21 ± 0.35	82.72 ± 0.26	72.05 ± 0.37	85.27 ± 0.25
Large Rotation	Task	**68.96 ± 0.35**	**83.65 ± 0.25**	**73.79 ± 0.36**	**85.81 ± 0.24**
CutMix	Task	68.78 ± 0.36	82.99 ± 0.50	72.72 ± 0.37	84.62 ± 0.25
Combining labels	Task	68.08 ± 0.35	82.33 ± 0.26	69.64 ± 0.37	83.79 ± 0.26
Random Erase	Task	68.39 ± 0.36	83.26 ± 0.25	71.09 ± 0.37	84.49 ± 0.25
Drop Channel	Task	67.54 ± 0.36	81.97 ± 0.25	70.24 ± 0.37	83.52 ± 0.26
Horizontal Flip	Shot	**68.13 ± 0.35**	82.95 ± 0.25	**73.25 ± 0.36**	**85.06 ± 0.25**
Random Crop	Shot	67.33 ± 0.36	**83.04 ± 0.25**	70.56 ± 0.37	83.87 ± 0.25
Random Rotation	Shot	67.57 ± 0.35	83.00 ± 0.25	70.32 ± 0.37	83.75 ± 0.25

Table 11. Few-shot classification accuracy (%) on the CIFAR-FS dataset with combinations of augmentations and query CutMix. "S","Q","T" denote "Support", "Query", and "Task" modes, respectively. While adding augmentations can help, it can also hurt, so additional augmentations must be chosen carefully.

	CNN-4		ResNet-12	
Mode	1-shot	5-shot	1-shot	5-shot
CutMix	70.54 ± 0.33	84.69 ± 0.24	75.97 ± 0.34	87.28 ± 0.23
+ CutMix (S)	69.50 ± 0.35	82.64 ± 0.26	75.00 ± 0.37	85.37 ± 0.25
+ Random Erase (S)	70.12 ± 0.35	84.48 ± 0.25	75.84 ± 0.34	87.19 ± 0.24
+ Random Erase (Q)	69.68 ± 0.34	84.36 ± 0.24	75.08 ± 0.35	87.14 ± 0.23
+ Self-Mix (S)	70.65 ± 0.34	84.68 ± 0.25	**76.27 ± 0.34**	87.52 ± 0.24
+ Self-Mix (Q)	69.94 ± 0.34	84.38 ± 0.24	76.04 ± 0.34	87.45 ± 0.24
+ MixUp (T)	70.33 ± 0.35	84.57 ± 0.25	75.97 ± 0.34	86.66 ± 0.24
+ Rotation (T)	70.35 ± 0.34	84.73 ± 0.24	75.74 ± 0.34	**87.68 ± 0.24**
+ Horizontal Flip (Shot)	**70.90 ± 0.33**	**84.87 ± 0.24**	76.23 ± 0.34	87.36 ± 0.24

F. Augmentation Pool for Meta-MaxUp

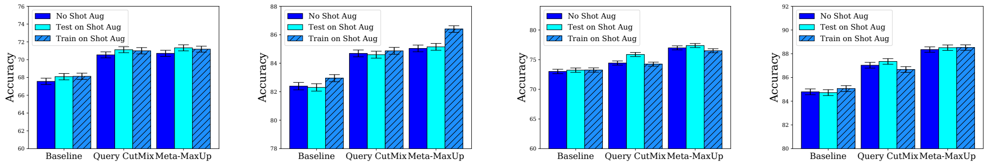
For all the benchmark results of Meta-MaxUp training, we use a medium-size data augmentation pool with $m = 4$, including CutMix (Q), Random Erase (Q), Self-Mix (S), Rotation (T), CutMix (Q) + Rotation (T), and Random Erase (Q) + Rotation (T). For the large-size pool, we add more techniques and combinations of the mentioned techniques into the pool, including Random Erase (Q) + Random Erase (S), CutMix (Q) + Random Erase (S), CutMix (Q) + Random Erase (Q), and CutMix (Q) + Self-Mix (S). Table 12 shows the few-shot classification accuracy on CIFAR-FS using R2D2 meta-learner and both CNN-4, ResNet-12 backbones, with various augmentations pool and hyper-parameter m.

Table 12. Few-shot classification accuracy (%) on the CIFAR-FS dataset for Meta-MaxUp over different sizes of augmentation pools and numbers of samples. As m and the pool size increase, so does performance. Meta-MaxUp is able to pick effective augmentations from a large pool.

		CNN-4		ResNet-12	
Pool	m	1-shot	5-shot	1-shot	5-shot
Baseline	-	67.56 ± 0.36	82.39 ± 0.26	71.95 ± 0.37	84.56 ± 0.25
CutMix	1	70.54 ± 0.34	84.69 ± 0.24	75.97 ± 0.34	87.28 ± 0.23
Single	1	70.76 ± 0.35	84.70 ± 0.25	75.71 ± 0.35	87.44 ± 0.43
Medium	1	70.50 ± 0.34	84.59 ± 0.24	75.60 ± 0.34	87.35 ± 0.23
Large	1	70.84 ± 0.34	85.04 ± 0.24	75.44 ± 0.34	87.47 ± 0.23
CutMix	2	70.56 ± 0.34	84.78 ± 0.24	74.93 ± 0.36	87.14 ± 0.24
Single	2	70.86 ± 0.34	85.06 ± 0.25	75.81 ± 0.34	87.33 ± 0.23
Medium	2	70.75 ± 0.34	85.02 ± 0.24	76.49 ± 0.33	88.20 ± 0.22
Large	2	70.63 ± 0.34	85.07 ± 0.24	76.59 ± 0.34	88.11 ± 0.23
CutMix	4	70.48 ± 0.34	84.76 ± 0.24	75.08 ± 0.23	87.60 ± 0.24
Single	4	**71.10 ± 0.34**	**85.50 ± 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	**76.99 ± 0.24**	**88.35 ± 0.22**

G. Bar Plots for Shot Augmentation

Figure 3 shows the effect of the shot augmentation in few-shot evaluation. In general, shot augmentation enhances the performance of meta-learners.



(a) (b) (c) (d)

Figure 3. Performance with shot augmentation using different backbones and training strategies on CIFAR-FS. (a) 1-shot classification with CNN-4 (b) 5-shot classification with CNN-4 (c) 1-shot classification with ResNet-12 (d) 5-shot classification with ResNet-12

H. Meta-Dataset Training and Evaluation

Details concerning each dataset in the Meta-Dataset benchmark can be found in Triantafillou et al. (2019). We use the same training procedure as mentioned above in Appendix C for both meta-learners. In each epoch, we train on 8000 episodes with shot of 5 and images of spacial dimensions 84 × 84 × 3, and we use mini-batches of 8 tasks each. When training with

Meta-MaxUp, we use the same augmentation pool as in Appendix F and set $m = 4$. During evaluation, we test 5-shot performance on 1000 tasks consisting of 15 query samples each. Due to the small number of sample size for several classes in the Fungi dataset, we use 1-shot classification with 5 query samples instead.