1 Implementation and Datasets details

1.1 Full MetaDataset Description

The MetaDataset includes `ImageNet` [12] (1000 categories of natural images), `Omniglot` [6] (1623 categories of black-and-white hand-written characters from different alphabets), `Aircraft` [9] (100 classes of aircraft types), `CU-Birds` [14] (200 different bird species), Describable `Textures` [1] (43 categories for textures), `Quick Draw` [3] (345 different categories of black-and-white sketches), `Fungi` [13] (1500 mushroom types), `VGG-Flower` [10] (102 flower species), `Traffic Sign` [2] (43 classes of traffic signs) and `MSCOCO` [7] (80 categories of day-to-day objects). For testing, we additionally employ `MNIST` [15] (10 hand-written digits) `CIFAR10` [5] (10 classes of common objects), and `CIFAR100` [5] (100 classes of common objects). Figure 1 illustrated random samples drawn from each dataset.

1.2 MetaDataset training details

When using multiple ResNet18 on MetaDataset (a single ResNet per dataset) to build a multi-domain representation, we train the networks according to the following procedure. For optimization, we use SGD with momentum and adjust the learning rate using cosine annealing [8]. The starting learning rate, the maximum number of training iterations ("Max iter.") and annealing frequency ("annealing freq.") are set individually for each dataset. To regularize training, we use data augmentation, such as random crops and random color augmentations, and set a constant weight decay of 7×10^{-4}. For each dataset, we run a grid search over batch size in [8, 16, 32, 64] and pick the one that maximizes accuracy on the validation set. The hyper-parameters maximizing the validation accuracy are given in Table 1.

When training a parametric network family for building multi-domain representations, we start by adopting a ResNet18 already trained on ImageNet, that we keep fixed for the rest of the training procedure. For each new dataset, we then train a set of domain-specific FiLM layers, modulating intermediate ResNet layers, as described in Section 3.3 of the original paper. Here, we also use cosine annealing as learning rate policy, employ weight decay and data augmentation, as specified above. In Table 2, we report the training hyper-parameters for each of the datasets.

1.3 mini-ImageNet training details

All the methods we evaluate on *mini*-ImageNet use ResNet12 [11] as a feature extractor. It is trained with batch size 200 for 48 epochs. For optimization, we use Adam optimizer [4] with initial learning rate 0.1 which is kept constant for the first 36 epochs. Between epochs 36 and 48, the learning rate was exponentially decreased from 0.1 to 10^{-5}, *i.e.* by dividing the learning rate by $10^{\frac{1}{3}}$ after each epoch. As regularization, we use weight decay with 5×10^{-4} multiplier and data augmentation such as random crops, flips and color transformations.

Test Dataset	learning rate	weight decay	Max iter.	annealing freq.	batch size
ImageNet	3×10^{-2}	7×10^{-4}	480,000	48,000	64
Omniglot	3×10^{-2}	7×10^{-4}	50,000	3,000	16
Aircraft	3×10^{-2}	7×10^{-4}	50,000	3,000	8
Birds	3×10^{-2}	7×10^{-4}	50,000	3,000	16
Textures	3×10^{-2}	7×10^{-4}	50,000	1,500	32
Quick Draw	1×10^{-2}	7×10^{-4}	480,000	48,000	64
Fungi	3×10^{-2}	7×10^{-4}	480,000	15,000	32
VGG Flower	3×10^{-2}	7×10^{-4}	50,000	1,500	8

Table 1: **Training hyper-parameters of individual feature networks on Meta-Dataset**. The first column indicates the dataset used for training. The first row gives the name of he hyper-parameter. The body of the table contains hyper-parameters that produced the most accurate model on the validation set.

Test Dataset	learning rate	weight decay	Max iter.	annealing freq.	batch size
Omniglot	3×10^{-2}	7×10^{-4}	40,000	3,000	16
Aircraft	1×10^{-2}	7×10^{-4}	30,000	1,500	32
Birds	3×10^{-2}	7×10^{-4}	30,000	1,500	16
Textures	3×10^{-2}	7×10^{-4}	40,000	1,500	16
Quick Draw	1×10^{-2}	7×10^{-4}	400,000	15,000	32
Fungi	1×10^{-2}	7×10^{-4}	400,000	15,000	32
VGG Flower	1×10^{-2}	7×10^{-4}	30,000	3,000	16

Table 2: **Training hyper-parameters of the parametric network family on MetaDataset**. The first column indicates the dataset used for training. The first row gives the name of he hyper-parameter. The body of the table contains hyper-parameters that produced the most accurate model on the validation set.

2 Additional Experiments and Ablation Study

2.1 Additional results on MetaDataset

Here we elaborate on using SUR with a multi-domain set of representations obtained from independent feature extractors (see Section 3.2), report an ablation study on varying the number of extractors in the multi-domain set, and report detailed results, corresponding to Figure 3 (a) of the original paper. Specifically, we use 8 domain-specific ResNet18 feature extractors to build a multi-domain representation and evaluate SUR against the baselines. The results are reported in Table 3, which corresponds to Figure 3 (a) of the original paper.

In the following experiment, we remove feature extractors trained on `Birds`, `Textures` and `VGG Flower` from the multi-domain feature set and test the performance of SUR on the set of remaining 5 feature extractors. We chose to remove these feature extractors as none of them gives the best performance on any of the test sets. Hence, they probably do not add new knowledge to the multi-domain set of features. The results are reported in Table 3 (a) as "SUR (5/8)". As we

Test Dataset	ImageNet-F	Union-F	Concat-F	SUR	SUR (5/8)
ImageNet	**56.3±1.0**	44.6±0.7	19.5±1.6	**56.0±1.2**	**56.1±1.1**
Omniglot	67.5±1.2	86.1±0.9	91.5±0.5	**93.0±0.4**	**93.1±0.4**
Aircraft	50.4±0.9	82.2±0.6	33.7±1.4	**85.7±0.3**	**85.5±0.4**
Birds	71.7±0.8	**72.1±1.1**	18.8±1.3	71.6±0.8	71.0±0.8
Textures	**70.2±0.7**	62.7±1.0	34.5±0.9	**70.3±0.9**	**70.4±0.9**
Quick Draw	52.3±1.0	70.7±0.9	51.2±0.9	**80.2±0.8**	**80.5±0.9**
Fungi	39.1±1.0	56.2±0.8	12.6±0.4	**62.8±1.1**	**63.1±1.0**
VGG Flower	**84.3±0.7**	82.5±0.8	40.3±1.2	83.6±0.8	83.3±0.8
Traffic Sign	63.1±0.8	63.8±0.9	48.2±0.6	**66.1±0.8**	63.6±0.9
MSCOCO	**52.8±1.0**	42.3±1.0	17.8±0.4	**52.4±1.1**	**52.8±1.1**
MNIST	77.2±0.7	84.8±0.6	89.6±0.7	91.2±0.5	**92.5±0.5**
CIFAR 10	**66.3±0.8**	51.4±0.8	34.7±0.8	64.6±0.9	65.8±0.9
CIFAR 100	**55.7±1.0**	39.5±1.0	18.9±0.6	54.5±1.0	**56.5±1.0**

Table 3: **Motivation for feature selection.** The table shows accuracy of different feature combinations on the Meta-Detaset test splits. The first column indicates the dataset the algorithms are tested on, the first row gives a name of a few-shot algorithm. The body of the table contains average accuracy and 95% confidence intervals computed over 600 few-shot tasks. The numbers in bold lie have intersecting confidence intervals with the most accurate method.

can see, selecting from the truncated set of features may be beneficial for some out-of-domain categories, which suggests that even the samples form of adaptation – selection – may overfit when very few samples are available. On the other hand, for a new dataset `Traffic Sign`, selecting from all features is beneficial. This result is not surprising, as one generally does not know what features will be useful for tasks not known beforehand, and thus removing seemingly useless features may result in a performance drop.

2.2 Analysis of Feature Selection on MetaDataset

Here, we repeat the experiment from Section 4.3, *i.e.* studying average values of selection parameters λ depending on the test dataset. Figure 2 reports the average selection parameters with corresponding confidence intervals. This is in contrast to Figure 4 of the original paper that reports the average values only, without confidence intervals.

2.3 Importance of Intermediate Layers on mini-ImageNet

We clarify the findings in Section 4.4 of the original paper and provide an ablation study on the importance of intermediate layers activations for the meta-testing performance. For all experiments on *mini*-ImageNet, we use ResNet12 as a feature extractor and construct a multi-domain feature set from activations of intermediate layers. In Table 4, we experiment with adding different layers outputs to the multi-domain set. The multi-domain set is then used to construct the final image representation either through concatenation "concat" or using

Method	1-3	4-6	7-9	10-12	Aggregation	5-shot	1-shot
Cls					last	76.28 ±0.41	60.09 ±0.61
				✓	select	77.39 ±0.42	**61.02 ±0.62**
			✓	✓	select	**79.25** ±0.41	60.79 ±0.62
		✓	✓	✓	select	78.92 ±0.41	60.71 ±0.64
	✓	✓	✓	✓	select	78.80 ±0.43	60.55 ±0.62
				✓	concat	78.43 ±0.42	60.41 ±0.62
			✓	✓	concat	75.67 ±0.41	57.15 ±0.61
		✓	✓	✓	concat	70.90 ±0.40	53.53 ±0.61
	✓	✓	✓	✓	concat	69.40 ±0.40	51.21 ±0.60
DenseCls					last	78.25 ±0.43	62.61 ±0.61
				✓	select	79.34 ±0.42	62.46 ±0.62
			✓	✓	select	**80.04** ±0.41	**63.13** ±0.62
		✓	✓	✓	select	79.84 ±0.42	62.95 ±0.62
	✓	✓	✓	✓	select	79.49 ±0.43	62.58 ±0.63
				✓	concat	79.12 ±0.41	62.51 ±0.62
			✓	✓	concat	79.59 ±0.42	62.74 ±0.61
		✓	✓	✓	concat	77.63 ±0.42	60.14 ±0.61
	✓	✓	✓	✓	concat	76.07 ±0.41	57.78 ±0.61
DivCoop					last	81.06 ±0.41	**64.14 ±0.62**
				✓	select	**81.23** ±0.42	63.83 ±0.62
			✓	✓	select	81.19 ±0.41	63.93 ±0.63
		✓	✓	✓	select	81.11 ±0.42	63.85 ±0.62
	✓	✓	✓	✓	select	81.08 ±0.42	63.71 ±0.62
				✓	concat	81.12 ±0.42	63.92 ±0.62
			✓	✓	concat	80.79 ±0.41	63.22 ±0.63
		✓	✓	✓	concat	80.52 ±0.42	62.48 ±0.61
	✓	✓	✓	✓	concat	80.36 ±0.42	61.30 ±0.61

Table 4: **Comparison to other methods on 1- and 5-shot *mini*-ImageNet.** The first column gives the name of the feature extractor. Columns 2-5 indicate if corresponding layers of ResNet12 were added to the multi-domain set of representations. Column "Aggregation" specifies how the multi-domain set was used to obtain a vector image representation. The two last columns display the accuracy on 1- and 5-shot learning tasks. To evaluate our methods we performed 1 000 independent experiments on *mini*-ImageNet-test and report the average and 95% confidence interval. The best accuracy is in bold.

SUR. The table suggests that adding the first 6 layers negatively influences the performance of the target task. While our SUR approach can still select relevant features from the full set of layers, the negative impact is especially pronounced for the "concat" baseline. This suggests that the first 6 layers do not contain useful for the test task information. For this reason, we do not include them in the multi-domain feature set, when reporting the results in Section 4.4.

We further provide analysis of selection coefficients assigned to different layers in Figure 3. We can see that for all methods, SUR picks from the last 6 layers most of the time. However, it can happen that some of the earlier layers are selected too. According to Table 4, these cases lead to a decrease in performance and suggest the SUR may overfit, when the number of samples if very low.

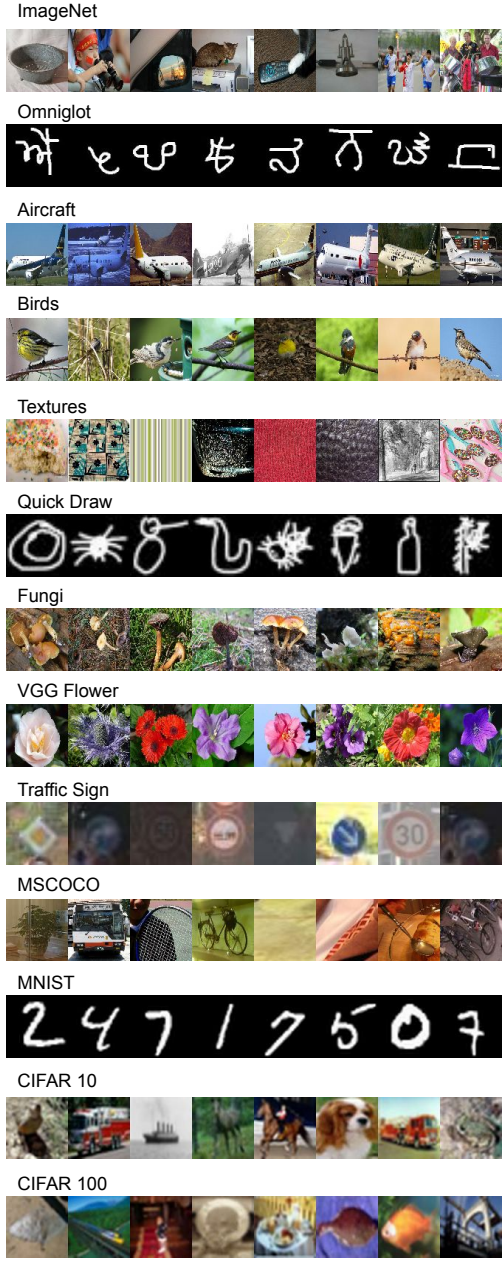

Fig. 1: **Samples from all MetaDataset datasets** Each line gives 8 random samples from a dataset specified above.

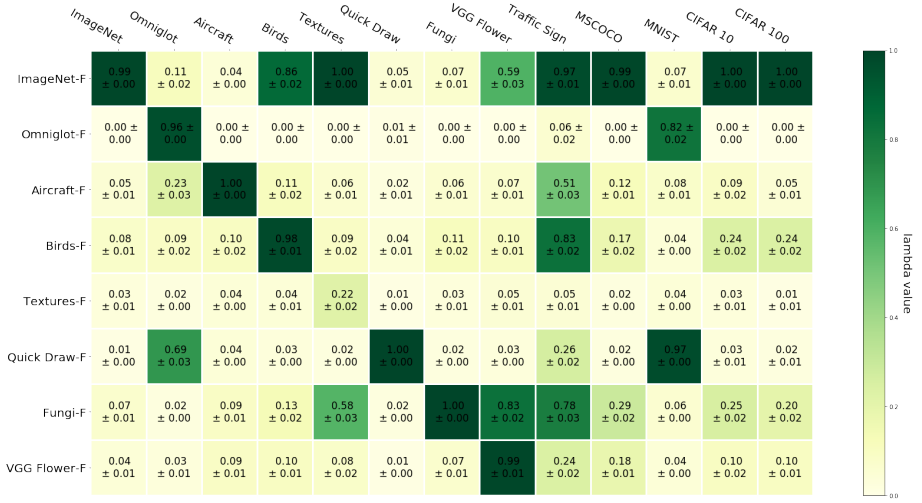

Fig. 2: **Frequency of selected features depending on the test domain in Meta-Dataset.** The top row indicates a testing dataset. The leftmost column presents a dataset the feature extractor has been trained on. A cells at location i, j reflects the average value of selection parameter λ_i assigned to the i-th feature extractor when tested on j-th dataset with corresponding 95% confidence intervals. The values are averaged over 600 few-shot test tasks for each dataset.

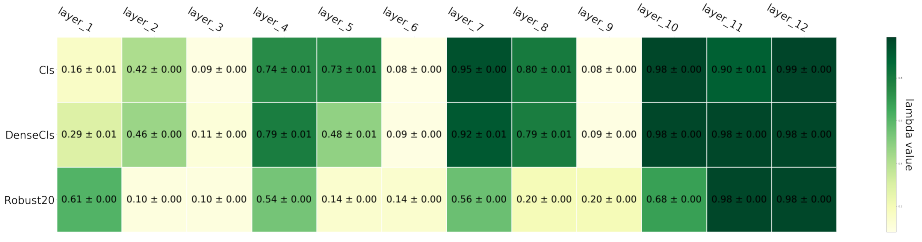

Fig. 3: **Frequency of selecting intermediate layer's activation's on mini-ImageNet for 5-shot classification.** The top row indicates intemediate layer. The leftmost column gives the name of a method used to pre-train the feature extractor. Each cells reflects the average value of selection parameter λ_i assigned to the i-th intemediate layer with corresponding 95% confidence intervals. The values are averaged over 1000 few-shot test tasks for each dataset.

References

1. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
2. Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: International Joint Conference on Neural Networks (IJCNN) (2013)
3. Jonas Jongejan, Henry Rowley, T.K.J.K., Fox-Gieg, N.: Fine-grained visual classification of aircraft. quickdraw.withgoogle.com (2016)
4. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
5. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
6. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science (2015)
7. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
8. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
9. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)
10. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Sixth Indian Conference on Computer Vision, Graphics & Image Processing (2008)
11. Oreshkin, B., López, P.R., Lacoste, A.: Tadam: Task dependent adaptive metric for improved few-shot learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Proceedings of the International Conference on Computer Vision (ICCV) (2015)
13. Schroeder, B., Cui, Y.: Fgvcx fungi classification challenge 2018. github.com/visipedia/fgvcx_fungi_comp (2018)
14. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
15. Yann LeCun, C.C., Burges, C.: Mnist handwritten digit database. http://yann.lecun.com/exdb/mnist (2010)